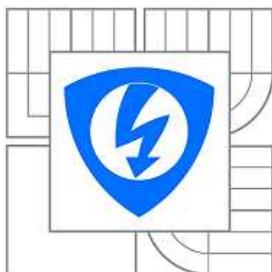


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE HRANIČNÍCH UZLŮ V BEZDRÁTOVÝCH SENZOROVÝCH SÍTÍCH

BOUNDARY NODES DETECTION IN WIRELESS SENSOR NETWORKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

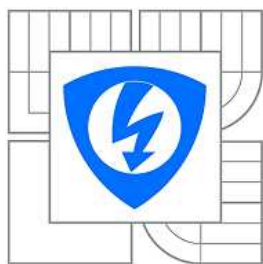
Bc. JAN BOCEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MILAN ŠIMEK, Ph. D.

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií
Ústav telekomunikací**

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Jan Bocek
Ročník: 2

ID: 98454
Akademický rok: 2010/2011

NÁZEV TÉMATU:

Detekce hraničních uzlů v bezdrátových senzorových sítích

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je výzkum algoritmů pro detekci hranic v bezdrátových senzorových sítích. Výstupem práce by měl být algoritmus, jehož redundance neboli počet špatně detekovaných hraničních uzlů bude minimální. Efektivitu navrženého algoritmu student porovná s algoritmy BRC a CGN. Algoritmus bude testován na sítích s různým počtem bezdrátových uzlů a s různou hustotou sítě. Výsledky budou reprezentovány jak formou grafů závislosti velikosti redundance na velikosti a hustotě sítě, tak pomocí zobrazených topologií s vyznačenými detekovanými uzly.

DOPORUČENÁ LITERATURA:

- [1] Stojmenovic I., Handbook of Sensor Networks, Wiley, ISBN: 13 978-0-471-68472-5, 2005.
- [2] Swami A., Zhao Q., Hong Y., Tong L., Wireless Sensor Networks, Signal Processing and Communications Perspectives, ISBN: 978-0-0470-03557-3, 2007.

Termín zadání: 7.2.2011

Termín odevzdání: 26.5.2011

Vedoucí práce: Ing. Milan Šimek, Ph.D.

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb

Abstrakt:

V teoretické části diplomové práce je popsáno několik známých algoritmů a nový přístup k detekci hraničních uzlů v bezdrátových senzorových sítích. Nový algoritmus BRB byl inspirován nedostatky předešlých algoritmů a jeho hlavní nosnou myšlenkou, je výběr několika sousedních bodů (B, C). Přístup algoritmu BRB je decentralizovaně detekovat uzly ležící na hranici sítě. Praktická část porovnává několik algoritmů s novým BRB. Vyhodnoceno je množství výskytu úspěšně a chybně určených hraničních uzlů v ideálním i reálném prostředí. Výsledky simulace dokázaly, že navržený algoritmus představuje efektivní řešení pro přesnou detekci hraničních uzlů v bezdrátových senzorových sítích.

Klíčová slova:

Bezdrátové senzorové sítě, detekce hranic sítě, senzor, výpočet vzdálenosti, reálné a ideální prostředí, simulace.

Abstract:

Theoretical part of thesis describes several known algorithms and new approaches for the border node detection in a wireless sensor networks. The new algorithm BRB was inspired by failures of previous algorithms and its major supporting idea is select several neighboring points (B, C). Approach algorithm BRB is a decentralized detection of nodes located at the perimeter. The practical part compares several algorithm with the new BRB algorithm. Quantity of successful and false detection of boundary nodes was evaluated in ideal and real environment. The simulation results shown, that the proposed algorithm is effective solution for accurate border node detection in wireless sensor networks.

Keywords:

Wireless Sensor Networks, Boundary Recognition, Sensor, Communication Cost, Real and Ideal Environment, Simulation

BOCEK, J. *Detekce hraničních uzlů v bezdrátových senzorových sítích*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 59 s. Vedoucí diplomové práce Ing. Milan Šimek, Ph.D..

Prohlášení

Prohlašuji, že svůj semestrální projekt na téma „Detekce hraničních uzlů v bezdrátových senzorových sítích“ jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 24.5.2011

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Milanu Šimkovi, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne 24.5.2011

.....

podpis autor

Obsah

Obsah.....	7
Seznam obrázků a tabulek	8
1. Úvod	9
2. Moderní bezdrátové senzorové sítě	10
2.1 Určení okrajových senzorů	10
2.2 Souběžné lokalizační algoritmy	11
2.3 Složitost algoritmu	13
3. Algoritmy pro detekci hraničních uzlů v WSN	15
3.1 Algoritmus CGN (Choose Good Neighbors).....	15
3.1.1. Předpoklady	15
3.1.2 Definice a poznámky.....	16
3.1.3 Rozbor problematiky.....	16
3.1.4 Návrhový algoritmus.....	17
3.1.5 Chyby algoritmu CGN	21
3.2 Algoritmus BRC (Boundary Recognition using Cset).....	24
3.2.1 Návrhový algoritmus.....	25
3.3 Algoritmus BRB (Boundary Recognition using Bset).....	28
3.3.1 Návrhový algoritmus.....	28
3.3.2 Shrnutí BRB	31
3.4 Algoritmus OBR (On boundary recognition without location information)	31
3.4.1 Úvod.....	31
3.4.2 Předpoklady	32
3.4.3 Hranice senzorové sítě	32
3.4.4 Vzory	33
3.4.5 Hranice rozpoznávání algoritmu	36
4. Analýza algoritmů pro detekci hraničních uzlů	37
4.1 Stručný popis simulačního programu	37
4.2 Rozbor grafů	38
4.3 Úspěšnost a chybovost detekce.....	43
4.4 Zhodnocení algoritmů CGN, BRC a BRB.....	48
4.4.1 Složitost algoritmu BRB	50
5. Chyby odhadu vzdálenosti.....	51
ZÁVĚR.....	53
SEZNAM LITERATURY	55
SEZNAM ZKRATEK, VELIČIN A SYMBOLŮ.....	57
PŘÍLOHA.....	59

Seznam obrázků a tabulek

Obr. 1. Základy AFL.....	12
Obr. 2. Vnitřní hranice a vnější hranice sítě.	16
Obr. 3. Volba správného trojúhelníku $\triangle ABC$, $\triangle ABD$	18
Obr. 4. Kroky pro výběr A , B , C , D algoritmu CGN.	19
Obr. 5. Situace umístění bodu P vzhledem k $\triangle ABC$, $\triangle ABD$	20
Obr. 6. Chyby algoritmu CGN – na stejné přímce.	22
Obr. 7. Chyby algoritmu CGN – nenalezeny body B , C	23
Obr. 8. Chyby algoritmu CGN – špatný výběr.	24
Obr. 9. Kroky pro výběr A , B , C_j algoritmu BRC.	25
Obr. 10. Rozdílné výsledky pro různé body B	30
Obr. 11. Problémy související s definicí hranic.	32
Obr. 12. a) Nevyhovující konstrukce pro vzor; b) a c) Vzory; d) Model.	33
Obr. 13. Kombinace tří chordless cyklů.	34
Obr. 14. Rozšíření nezávislého souboru vlastností.	34
Obr. 15. Test zbarvení a typy kritických průsečíků.	35
Obr. 16. Rozšířené vzory.	36
Obr. 17. CGN, BRC, BRB sítě s 50 uzly modely 3. a 9.	39
Obr. 18. CGN, BRC, BRB sítě se 100 uzly modely 3. a 9.	40
Obr. 19. CGN, BRC, BRB sítě se 400 uzly modely 3. a 9.	41
Obr. 20. Umístění bodu P v síti.....	43
Obr. 21. Příklad Box plotu.....	44
Obr. 22. Úspěšnost a chybovost detekce sítí s 50 uzly.	45
Obr. 23. Úspěšnost a chybovost detekce sítí se 100 uzly.	46
Obr. 24. Úspěšnost a chybovost detekce sítí se 400 uzly.	47
Obr. 25. Úspěšnost a chybovost detekce sítí s 50 uzly v závislosti na hustotě sítě.	47
Obr. 26. Úspěšnost a chybovost detekce sítí se 100 uzly v závislosti na hustotě sítě.	48
Obr. 27. Úspěšnost a chybovost detekce sítí se 400 uzly v závislosti na hustotě sítě.	48
Obr. 28. Úspěšnost a chybovost detekce v závislosti na chybě odhadu vzdálenosti sítě.	52
Tab. 1. Nejčastější třídy složitosti.....	13
Tab. 2. Pseudokód algoritmu CGN (Choose good neighbors).	18
Tab. 3. Pseudokód algoritmu IP (Interior Point).	20
Tab. 4. Pseudokód algoritmu BRC část createCset.	26
Tab. 5. Pseudokód algoritmu BRC část boundaryRecognition.	27
Tab. 6. Pseudokód algoritmu BRB část createBset (n_i).	29
Tab. 7. Pseudokód algoritmu BRB část boundaryRecognition (P).	31
Tab. 8. Parametry simulace algoritmu CGN, BRC a BRB, ideální prostředí.	38
Tab. 9. Tabulka hodnot CGN, BRC, BRB.	45
Tab. 10. Parametry simulace algoritmů CGN, BRC a BRB, reálné prostředí.	51

1. Úvod

Cílem diplomové práce je výzkum algoritmů pro detekci hraničních senzorů v bezdrátových sensorových sítích. Výstupem práce by měl být algoritmus, který dosahuje minimálních hodnot redundance neboli počet chybně detekovaných hraničních uzlů. Jelikož většina známých algoritmů detekuje téměř všechny hraniční uzly, je podstatnějším hlediskem snížit redundanci na minimum. Zmenšením redundance dojde u zbylých uzlů (ne hraničních) k ušetření energie baterie, a tím se i prodlouží délka života uzlu. Tedy vhodným způsobem vybrat sousední uzly pro dosažení větší přesnosti detekce. Navržený algoritmus bude porovnán se známými algoritmy BRC a CGN v ideálním i reálném prostředí. Algoritmy budou testovány na sítích s různým počtem bezdrátových uzlů a s různou hustotou sítě. Jako srovnávací faktor poslouží jak velikosti úspěšnosti a chybovosti detekce na velikosti a hustotě sítě, tak pomocí zobrazených topologií s vyznačenými detekovanými uzly.

Práce obsahuje tyto hlavní kapitoly: Úvod, Moderní bezdrátové sensorové sítě, Algoritmy pro detekci hraničních uzlů v WSN, Analýza algoritmů pro detekci hraničních uzlů, Chyby odhadu vzdálenosti a Závěr. Druhá kapitola se zabývá obecným vysvětlením, co je síť senzorů, využití v praxi, složitostmi algoritmů, proč je potřeba určit hraniční uzly a jak tyto uzly určit bez informací o poloze vzhledem ke globálnímu souřadnicovému systému. Třetí kapitola popisuje princip algoritmů CGN, BRC, BRB a OBR pro určení hraničních uzlů. Rovněž rozebírá některá úskalí a nedokonalosti algoritmů. Ve čtvrté kapitole je stručně popsán postup simulačního programu, zobrazené rozložení úspěšně a chybně detekovaných hraničních uzlů v sítích simulovaných algoritmů CGN, BRC a BRB. Tyto výsledky jsou pak zpracovány do grafů pomocí úspěšnosti a chybovosti detekce. Dále jsou porovnány grafy závislosti úspěšnosti a chybovosti detekce na hustotě sítě. Na konci kapitoly jsou detailně rozebrány výhody a nevýhody jednotlivých simulovaných algoritmů. V páté kapitole jsou algoritmy simulovány v reálném prostředí a výsledky jsou vyhodnoceny jako závislost úspěšnosti a chybovosti detekce na chybě odhadu vzdálenosti sítě.

2. Moderní bezdrátové senzorové sítě

Bezdrátové senzorové sítě (WSN) se skládají z prostorově distribuovaných autonomních senzorů pro sledování fyzické nebo environmentální situace, jako je teplota, zvuk, vibrace, tlak, pohyb nebo znečišťující látky. O rozvoj bezdrátových senzorových sítí se postarala především armáda a až později se uplatňovaly v průmyslové a civilní oblasti. Původně měly dohlížet na bojišti a zlepšit techniku boje a nyní jsou uplatněny k monitorování a řízení procesů, sledování zdravotního stavu, životní prostředí či přírodních katastrof. [13]

Velké množství dnešních senzorových sítí je tvořeno desítkami, stovkami levných zařízení (senzorů), které jsou propojeny nízkou výkonovou bezdrátovou komunikací definovanou standardem IEEE 802.15.4. Aktivní uzly senzorové sítě mezi sebou spolupracují a předávají si užitečné informace. Pokroky v posledních desetiletích zaznamenala mikroelektronika, snímače, analogové i digitální zpracování signálu, bezdrátová komunikace v sítích a technologie bezdrátových senzorových sítí. Očekává se, že tento pokrok bude mít významný dopad na životy lidí. Využití je téměř neomezené, od monitorování životního prostředí, predikce nebo snímání průběhu přírodních katastrof, vlastního zabezpečení, domácích spotřebičů, výroby, zdravotnictví, po řízení dopravy, vojenské mise či kosmonautiku. [13]

Bezdrátové senzorové sítě se zásadně liší od obecných datových sítí, jako je internet, a jako takové vyžadují přijetí odlišné konstrukce paradigmat (určitý vzor vztahů či vzorec myšlení). Ve většině případů jsou senzorové sítě specifickými aplikacemi navrženými a určenými pro speciální účely využití v reálném prostředí. Při návrhu sítě je zapotřebí vzít v úvahu, jak specifické použití, tak i povahu vysílání média. Pro senzory napájené bateriemi je úspora energie jedním z nejdůležitějších konstrukčních parametrů, protože výměna baterie může být často velmi obtížná nebo i nemožná. [1] O problematice spotřeby a úspory energie pojednává literatura. [7]

2.1 Určení okrajových senzorů

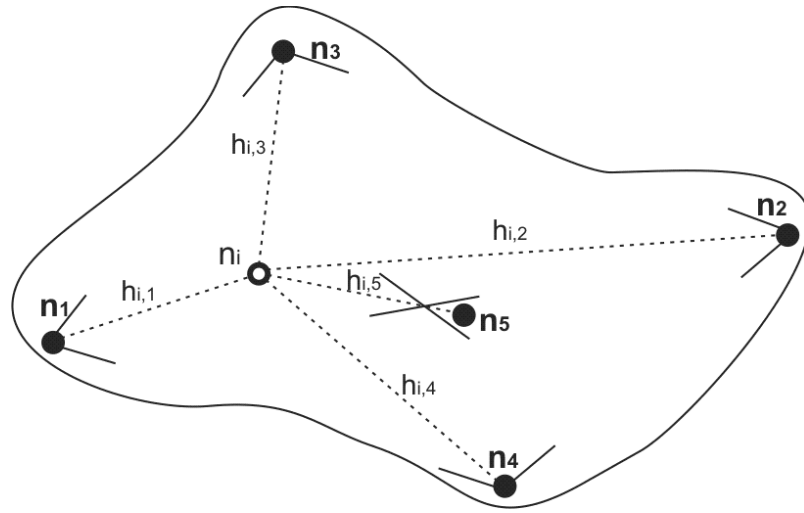
Desítky, stovky senzorů se mohou rozprostírat na určitém území, kde jsou náhodně či pravidelně rozmístěny, a všechny spolu navzájem komunikují. Každá komunikace zkracuje dobu života senzoru, a pokud aplikace vyžaduje snímání informace pouze z určitých senzorů

(např. krajních), není zapotřebí, aby byly všechny v aktivním módu (Transmit, Receive). Aplikace může často vyžadovat informace pouze od okrajových senzorů a z toho důvodu je zapotřebí určit tyto senzory a spustit aktivní mód. U zbylých uzlů se ušetří energie baterie, a tím se i prodlouží délka života uzlu. Algoritmus pro souběžnou lokalizaci okrajových uzlů je popsán v kapitole 2.2 Souběžné lokalizační algoritmy. Nejsou-li uzly vybaveny GPS (Global Position System) přijímačem, znamená to, že nejsou schopny určit svou polohu vzhledem ke globálnímu souřadnicovému systému, a tak i vzhledem k ostatním uzlům. Častým důvodem, proč senzory nejsou vybaveny GPS přijímačem, je cena, která se pohybuje řádově v jednotkách tisíců Kč. U některých aplikací jsou vyžadovány malé a levné senzory, a proto nelze GPS přijímač využít. I přesto lze na základě síly signálu určit seznam sousedních uzlů a jejich vzdálenost. Pomocí určitých algoritmů popsaných níže (viz kapitola 3. Algoritmy pro detekci hraničních uzlů v WSN) lze určit okrajové uzly a tuto informaci využít např. pro monitorování cizích objektů (nepřátel) na bitevním poli. V tomto případě by krajní uzly monitorovaly změny teploty, tlaku, zvuku, světla, elektro-magnetického pole či vibrací. Informace by byla okamžitě hlášena základnové stanici. V případě vybití akumulátoru, chyby hardwaru, přemístění či odcizení senzoru je zapotřebí znova určit okrajové senzory. [5]

2.2 Souběžné lokalizační algoritmy

Pro potřebu výběru pouze hraničních uzlů v síti z důvodu celkové úspory energie byl navržen algoritmus pro souběžnou lokalizaci okrajových uzlů. U zbylých uzlů se ušetří energie baterie, a tím se i prodlouží délka života uzlu.

Všechny aktivní uzly provozují lokalizační algoritmus nezávisle na ostatních uzlech v síti. Šíření chyb je tedy potlačeno na minimum, protože se uzly nespolehlí na dříve stanovenou pozici uzlů. K tomuto účelu byl navržen algoritmus nazvaný AFL (Anchor-Free Localization), který využívá pět uzlů působících jako referenční soubor pro vytvoření volného uspořádání. [6]



Obr. 1. Základy AFL.

Hlavní výběrová politika definuje čtyři uzly jako $n_1 \dots n_4$, které mají být umístěny na okraji sítě viz Obr. 1. Dále by měly být umístěny co nejdále od sebe, zhruba v okrajích pomyslného čtyřúhelníku. Okraj n_1, n_2 by měl být přibližně kolmo k okraji n_3, n_4 . Pátý uzel n_5 by měl být co nejvíce v oblasti těžiště sítě. Náhodně vybraný uzel zahájí fázi výběru tím, že hledá nejdále umístěný uzel. Tento úkol lze provést vysláním všesměrové zprávy `DISTANCE_QUERY` náhodně vybraným uzlem n_i (iniciačním uzlem). Každý uzel, který tuto zprávu obdrží, uloží počet přeskoků že zpráva prošla a inkrementuje o 1. Vzhledem k charakteru komunikace vysílání může každý uzel získat více kopií zprávy `DISTANCE_QUERY`. Pokud uzel obdrží zprávu obsahující menší počet přeskoků h , než již uložil, původní počet h se přepíše a zpráva je přeposlána dále. Jakmile všechny uzly znají nejkratší cestu k inicializačnímu uzlu, pošlou zpátky inicializačnímu uzlu zprávu `DISTANCE_RESPONSE` obsahující informaci $(ID|h)$. Inicializační uzel porovná vzdálenosti od všech uzlů a oznámí výběr uzlu n_1 s maximální h . Takto je vybrán první referenční uzel. Tento uzel pak pokračuje ve fázi referenčního výběru. Uzel n_1 vybere uzel n_2 a uzel n_3 je zvolen ve spolupráci uzlů n_1 a n_2 jako uzel se stejnou vzdáleností od obou uzlů. Následně uzel n_3 vybere uzel n_4 . Uzel n_5 musí splňovat podmínku, aby byl zhruba ve stejné vzdálenosti od hran všech uzlů n_1, n_2, n_3, n_4 . Jakmile je stanovena referenční sada, každý uzel zná vzdálenosti všech okrajových uzlů a jednoho uzlu n_5 uprostřed sítě. Na základě referenční sady pak vypočítá vlastní polární souřadnice jako: [6]

$$\rho_i = h_{i,5} \times R \quad (2.1)$$

$$\theta_i = \tan^{-1} \left(\frac{h_{i,1} - h_{i,2}}{h_{i,3} - h_{i,4}} \right) \quad (2.2)$$

2.3 Složitost algoritmu

Algoritmus lze definovat jako konečný počet elementárních kroků vedoucí k řešení určitého problému. (s rostoucím počtem kroků roste složitost algoritmu). Složitost algoritmu se klasifikuje podle doby prováděného algoritmu (časová složitost) a rozsahu použité operační paměti (paměťová složitost). Jelikož složitost závisí také na velikosti vstupních dat, lze ji popsat funkcí $T(n)$, kde n je velikost vstupních dat. Lineární časová složitost je pak např.

$$T(n) = a * n + b \quad (2.3)$$

,kde a, b jsou libovolné konstanty. Složitost roste lineárně s rostoucí velikostí vstupních dat. Pro odhad složitosti je obvykle důležitý pouze typ funkční závislosti a přesné vstupní hodnoty se mohou zanedbat. [8] V Tab. 1 [9] jsou uvedeny nejčastější třídy složitosti ($O(n)$ – Notace Omikron) použité při popisu algoritmů.

Notace	Název	Příklad
$O(1)$	konstantní	indexování prvků v poli
$O(\log n)$, $n \in \langle 0, \infty \rangle$	logaritmická	Nalezení prvku v ADT binárním vyhledávacím stromě [10]
$O(n)$	lineární	Nalezení prvku v neseřazeném poli
$O(n \log n)$	kvazilineární	Seřazení seznamu pomocí Heap sort [12], výpočet FFT
$O(n^2)$	kvadratická	Řazení pomocí Insertion sort [12], výpočet DFT
$O(n^3)$	kubická	Řazení pomocí Bubble sort (obdobné příklady jako kvadratická složitost) [12]
$O(c^n), c > 1$	exponenciální	Přesné řešení problému obchodního cestujícího [11]
$O(n!)$	faktoriální	Řešení NP úplného problému [11] hrubou silou (problém obchodního cestujícího)

Tab. 1. Nejčastější třídy složitosti.

Algoritmus lze považovat za efektivní, pokud jeho složitost je polynomiální (např. n^{127} , nikoliv exponenciální 2^n). Provádění exponenciální či faktoriální složitosti algoritmu může už při malém navýšení velikosti vstupních dat trvat velmi dlouhý čas (měsíce, roky). Jelikož nelze vždy určit přesnou složitost algoritmu, byla zavedena asymptotická složitost. Asymptotická

složitost aproximuje chování funkce z pohledu nejlepšího možného chování (dolní odhad složitosti), průměrného chování (očekávaná složitost) či nejhoršího chování (horní odhad složitosti). Očekávaná složitost se počítá jako střední hodnota složitosti při náhodném rozložení vstupních dat. Horní odhad složitosti udává složitost algoritmu v nejhorším případě. Někdy ovšem algoritmus dosahuje této horní hranice složitosti jen ve vzácných případech. V takových případech o složitosti algoritmu lépe vypovídá složitost průměrná (očekávaná). [8]

3. Algoritmy pro detekci hraničních uzlů v WSN

Pro určení okrajových uzlů existuje několik algoritmů. V následujícím textu budou podrobně rozebrány známé algoritmy CGN (Choose Good Neighbors) viz kapitola 3.1, BRC (Boundary Recognition using Cset) viz kapitola 3.2 a algoritmus OBR (On boundary recognition without location information) viz kapitola 3.4. Dále je také v kapitole 3.3 popsán nový algoritmus BRB (Boundary Recognition using Bset).

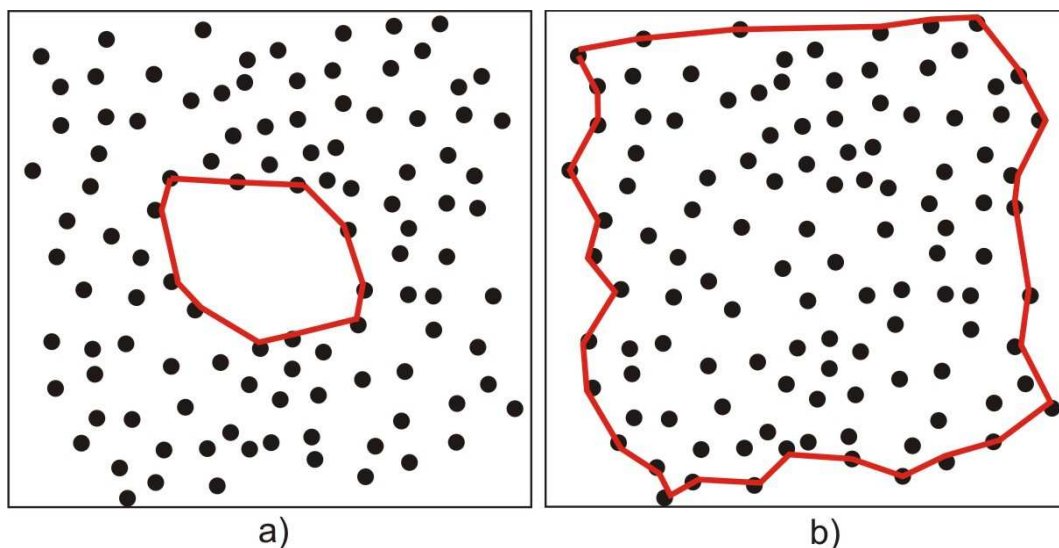
3.1 Algoritmus CGN (*Choose Good Neighbors*)

3.1.1. Předpoklady

Celou síť WSN si lze představit jako náhodně rozmístěné uzly na určité ploše. Nemusí být vybavený GPS, ale i přesto lze určit vzdálenost k jejich sousedním uzlům, a to na základě síly signálu. Předpokladem pro lokalizaci je, že každý z uzlů má dosah na minimálně 3 sousední uzly a nejedná se o síť s vnitřní hranicí viz Obr. 2. a).

Předpoklady lze shrnout do několika bodů.

1. Každý uzel je schopen určit vzdálenost od kteréhokoliv ze sousedních uzlů v dosahu signálu.
2. Každý uzel v síti má unikátní umístění, které ovšem není známé.
3. Síť je dostatečně hustá tak, aby každý uzel v síti měl nejméně 3 sousední uzly ve svém dosahu, z nichž lze vypočítat vzájemnou vzdálenost.
4. Vnitřní hranice sítě. V síti se nevyskytují žádné oblasti ohraničené uzly, kde by se uprostřed nenacházely další uzly viz Obr. 2. a).
5. Komunikace mezi uzly je obousměrná a symetrická. [2]



Obr. 2. Vnitřní hranice a vnější hranice sítě.

3.1.2 Definice a poznámky

Senzorová síť s n uzly je možné modelovat jako graf $G = (V, E)$, kde $n = V$ představuje soubor uzlů v síti, a E je množina neorientovaných hran $e(i, j)$. Platí že $i, j \in V$ a uzel j je v rámci komunikačního dosahu uzlu i . Necht' $P \in V$ je uzel v síti, definuje $N^k(P)$ patřící do souboru všech uzlu, které jsou uvnitř k skoků od uzlu P . Formálně, $N^k(P) = \{i : i \in V, d(P, i) \leq k\}$, kde $d(P, i)$ představuje vzdálenost mezi uzly i a P . Proto $N^k(P)$, představuje k^{th} skupinu uzavřených sousedních uzlů P v G . Soubor $N^1(P)$ pak obsahuje všechny uzly, které jsou v dosahu komunikace uzlu P . Formálně $N(P) = \{i : i \in V, (P, i) \in E\}$.

Definice 1 (Vnitřní uzly). Uzel $P \in V$ je *vnitřní* uzel, pokud E a další 3 uzly $A, B, C \in V$ jsou takové, že $A, B, C \in N(P)$ a uzly A, B, C tvoří trojúhelník obklopující uzel P .

Definice 2 (Okrajové uzly). Uzel $P \in V$ je *hraniční* uzel, pokud P není *vnitřní* uzel. Definujeme B jako hraniční uzel v síti a I bude nastaven jako vnitřní, tzn. ne hraniční.

Definice 3 (Okrajové spoje). Pomyslná čára, která spojuje hraniční (okrajové) uzly sítě senzorů je definovaná jako síťové rozhraní. Síťové rozhraní definuje obvod celé sítě. [2]

3.1.3 Rozbor problematiky

Vzhledem k tomu, že se jedná o nahodilou distribuovanou síť s několika uzly a neznámými souřadnicemi, je zapotřebí určit mechanismus pro určení jejich vzdáleností

od svých sousedů. Výsledným cílem je pak určit hraniční uzly sítě.

Problematiku lze formálně určit takto. Vzhledem k tomu, že graf $G = (V, E)$, kde uzly grafu G jsou vloženy do 2D prostoru a pro uzel $p \in V$, je zapotřebí zjistit, zda existuje soubor S a platilo $S \in N(p)$ a $|S| \geq 3$. Tedy existují nejméně tři uzly $a, b, c \in S$ a uzel p je uzavřen uvnitř trojúhelníku. [2]

3.1.4 Návrhový algoritmus

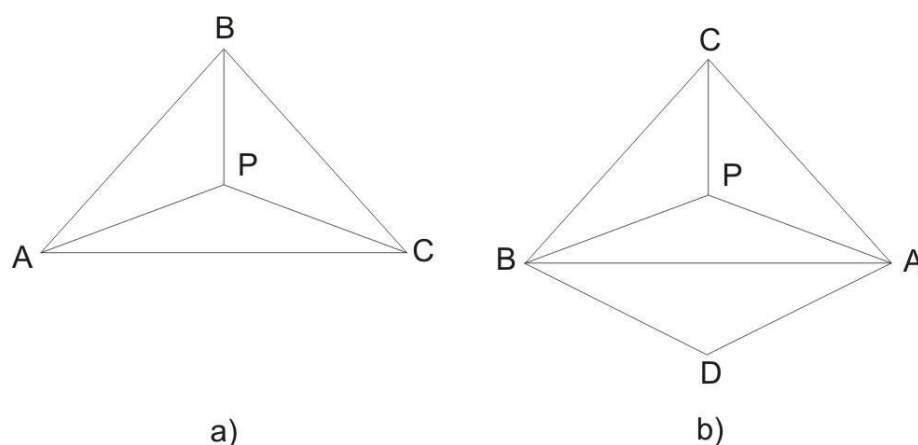
V této sekci je popsán navržený algoritmus pro detekci hranic sítě tedy CGN (Choose Good Neighbors). Algoritmus má za úkol vybrat čtyři vhodné sousedy A, B, C a D pro uzel P.

<u>CGN (Choose Good Neighbors)</u>	
Vstup:	seznam sousedních bodů pro P
Výstupy:	Čtyři vhodné sousední uzly k uzlu P. tzn. A, B, C, D.
1.	for $\forall n \in N(P)$ do Výběr bodu A s $\min(d_{PA})$ end
2.	for $\forall n \in N(P) \ \& \ N(A)$ do Výběr bodu B s $\max(d_{AB})$ end
3.	for $\forall n \in N(P) \ \& \ N(A) \ \& \ N(B)$ do Výběr bodu C s $\min(d_{AC} - d_{BC}) \ \& \ \min(d_{PC})$ end
4.	for $\forall n \in N(P) \ \& \ N(A) \ \& \ N(B) \ \& \ N(C)$ do Výběr bodu D s $\min(d_{AD} - d_{BC}) \ \& \ \max(d_{DC}) \ \& \ d_{DC} > d_{DP} \ \& \ d_{DC} > d_{PC}$ end

5.	<pre> if (InteriorPoint (P, A, B, C)==pravda) return 'P je vnitřní bod.' elseif (InteriorPoint (P, A, B, D)==pravda) return 'P je vnitřní bod.' else return 'P je hraniční bod.' [2] </pre>
----	---

Tab. 2. Pseudokód algoritmu CGN (Choose good neighbors).

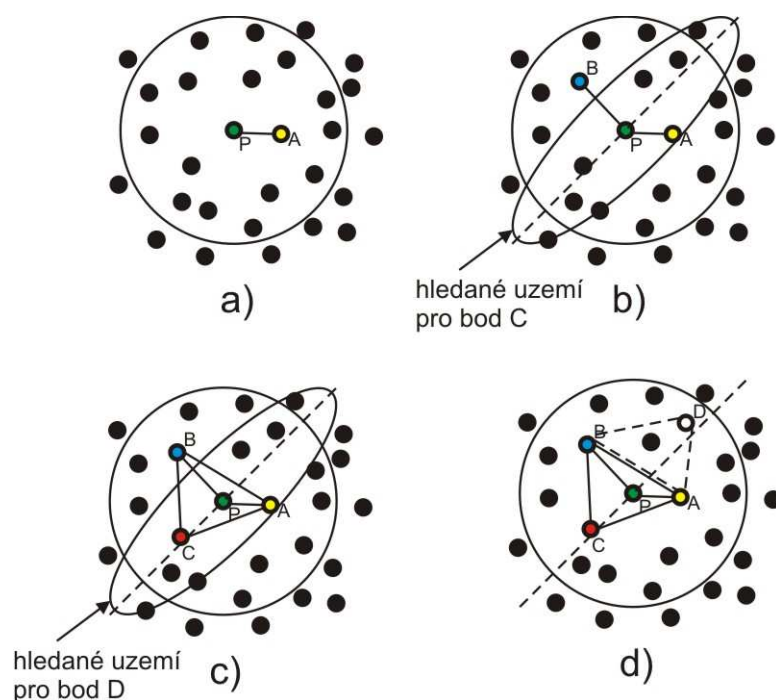
Algoritmus CGN slouží k výběru vhodných sousedních bodů k bodu P. Dále pak algoritmus IP určí, zda je tento bod uvnitř, či vně. Je vhodné poznamenat, že přesnost tohoto algoritmu závisí na tom, jak jsou vybrány tři sousední body A, B, C uzlu P. Jelikož každý uzel P může mít i více než tři sousední uzly, je možné že P bude vnitřní nebo vnější v závislosti na tom, jak zvolíme tyto sousedy. Situace je znázorněna na Obr. 3. Jsou-li vybrány sousední uzly A, B a C, pak P bude vnitřní, tzn. je uzavřen uvnitř $\triangle ABC$ (Obr. 3. a)). Ovšem pokud zvolíme $\triangle ABD$, pak bod P je vnější uzel (Obr. 3. b)). [2]



Obr. 3. Volba správného trojúhelníku $\triangle ABC$, $\triangle ABD$.

Obr. 4. vysvětluje, jak algoritmus CGN funguje a jak vybírá sousední uzly k danému uzlu P. První hledaný bod (A) má nejmenší vzdálenost od uzlu P viz Obr. 4. a). Bod B je hledán tak, že vzdálenost AB je maximální, ovšem musí ležet v $N(P)$ i $N(A)$. viz Obr. 4. b). Třetí bod C je hledán ve stejné vzdálenosti od bodu A i B, ovšem v minimální od bodu P. Nacházet se může kdekoli uvnitř hledaného území pro bod C podle Obr. 4. c). Nachází-li se C

na opačné straně než A, B viz Obr. 4. c), pak to znamená, že bod P leží uvnitř $\triangle ABC$. Pokud by bod P neležel uvnitř $\triangle ABC$ je zapotřebí najít D, který bude opět ve stejné vzdálenosti od bodu A i B a v maximální od C. Hledaný bod D by měl splňovat podmínky stanovené v kroku 4. pro CGN. Pokud bod D existuje, je zapotřebí, aby se nacházel na opačné straně než C Obr. 4. d). Není-li bod D nalezen a současně P není uvnitř $\triangle ABC$, pak P je hraniční uzel. [2]



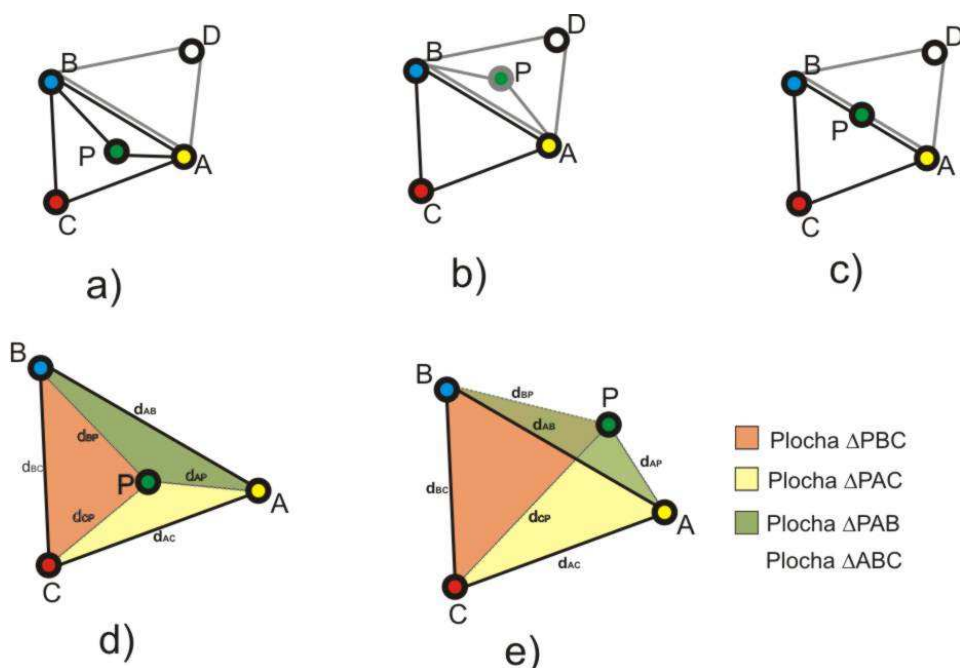
Obr. 4. Kroky pro výběr A, B, C, D algoritmu CGN.

Další částí algoritmu CGN je část IP (Interior Point) viz pseudokód Tab. 3. IP algoritmus navazuje na CGN, lépe řečeno dále pracuje a vyhodnocuje výsledky. Z vhodně vybraných bodů A, B, C a D určí, zda daný bod P je vnitřní bod trojúhelníků $\triangle ABC$ nebo $\triangle ABD$. Níže je popsán postup IP pro případ s $\triangle ABC$, pro trojúhelník ABD platí obdoba, ovšem s odlišnými vstupními proměnnými. [2]

IP (Interior Point)	
Vstup:	Čtyři body P, A, B, C
Výstupy:	InteriorPoint = pravda/nepravda
1.	Výpočet obsahů trojúhelníků $\triangle ABC, \triangle PAC, \triangle PBC$ a $\triangle PAB$ dle Heronova

	vzorce. $Plocha = \sqrt{[s(s-a)(s-b)(s-c)]}$,kde $s = (a + b + c) / 2$ a a, b, c jsou délky tří stran trojúhelníka Δ .
2.	if $Plocha \Delta ABC = Plocha (\Delta PAC + \Delta PBC + \Delta PAB)$ return InteriorPoint =pravda else return InteriorPoint =nepravda [2]

Tab. 3. Pseudokód algoritmu IP (Interior Point).



Obr. 5. Situace umístění bodu P vzhledem k ΔABC , ΔABD .

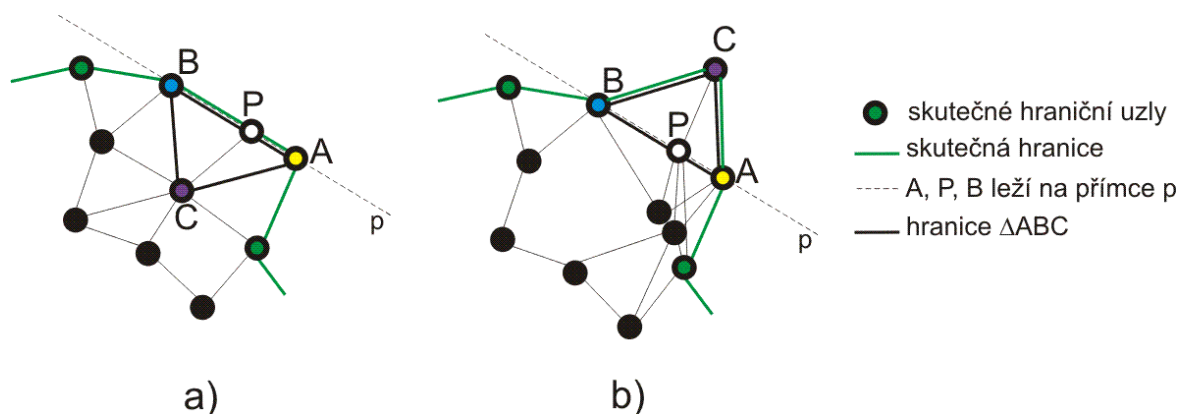
Na Obr. 5. jsou znázorněny různé možné situace výskytu bodu P vzhledem k oběma trojúhelníkům. První situace viz Obr. 5. a) zobrazuje bod P ležící uvnitř trojúhelníku ΔABC a neležící uvnitř ΔABD . Podmínka pro výběr bodu D v algoritmu CGN ošetřuje případ, že by ΔABC byl podmnožinou ΔABD , a tím pádem by bod P ležel v obou trojúhelnících, a proto tato situace nemůže nastat. Druhá situace nastane, pokud bod P není uvnitř trojúhelníku ΔABC . Pak je podstoupen výpočet, zda P je uvnitř ΔABD viz Obr. 5. b). Obr. 5. c) znázorňuje situaci, kdy se P nachází na hraně AB, a tím pádem nastává problém, zda leží uvnitř ΔABC nebo ΔABD . Pokud je nalezen vhodný bod D tak P bude vyhodnocen jako vnitřní ΔABC . I kdyby byl bod P vyhodnocen jako ležící uvnitř ΔABD nebylo by to směrodatné, protože

hlavním úsilím algoritmu je nalézt skutečné hraniční uzly. Skutečné hraniční uzly se mohou lišit od uzlů ležících vně $\triangle ABC$ a $\triangle ABD$ viz Obr. 20. a). Pokud ovšem není nalezen vhodný bod D, pak P je vyhodnocen jako vnitřní, a přitom by se mohlo jednat o skutečný hraniční bod viz Obr. 5. a). Obr. 5. d) ilustruje dílčí plochy trojúhelníků $\triangle PBC$, $\triangle PAC$, $\triangle PAB$. Rozhodující podmínka (IF $Plocha \triangle ABC = Plocha (\triangle PAC + \triangle PBC + \triangle PAB)$) v bodě 2., vyhodnotí celkovou plochu $\triangle ABC$ rovnou součtů dílčích ploch $\triangle PBC$, $\triangle PAC$, $\triangle PAB$ tzn. bod P je uvnitř $\triangle ABC$.

3.1.5 Chyby algoritmu CGN

Použití algoritmu CGN je inspirativní z hlediska toho, že každému uzlu P jsou vybrány vhodné sousední uzly pro vytvoření $\triangle ABC$ a $\triangle ABD$, ke kterým je pak určen vztah vnější, vnitřní. I tento algoritmus má své nedokonalosti, proto budou v této kapitole rozebrány některé z nich.

- 1) První z nedokonalostí návrhu je rozhodovací podmínka $\triangle ABC = Plocha (\triangle PAC + \triangle PBC + \triangle PAB)$. Za předpokladu, že nebyl nalezen vhodný bod D, IP algoritmus rozhodne pouze na základě $\triangle ABC$, a tím pádem může nastat případ na Obr. 6. a). Body A, P, B jsou na jedné přímce a plocha trojúhelníku $\triangle PAB$ je rovna 0. Pak součet ploch $\triangle PAC + \triangle PBC$ je roven ploše $\triangle ABC$, tedy podle rozhodovací podmínky (Tab. 3. bod 2.) uzel P je vnitřní bod. Pokud by rozhodovací podmínka byla ošetřena předpokladem, bude-li libovolná plocha dílčích trojúhelníků ($\triangle PAC$, $\triangle PBC$ či $\triangle PAB$) nulová, pak P je vnější. To je ovšem v rozporu s Obr. 6. b), kde bod P je vnitřním uzlem. Proto tento problém má nejednoznačné řešení.

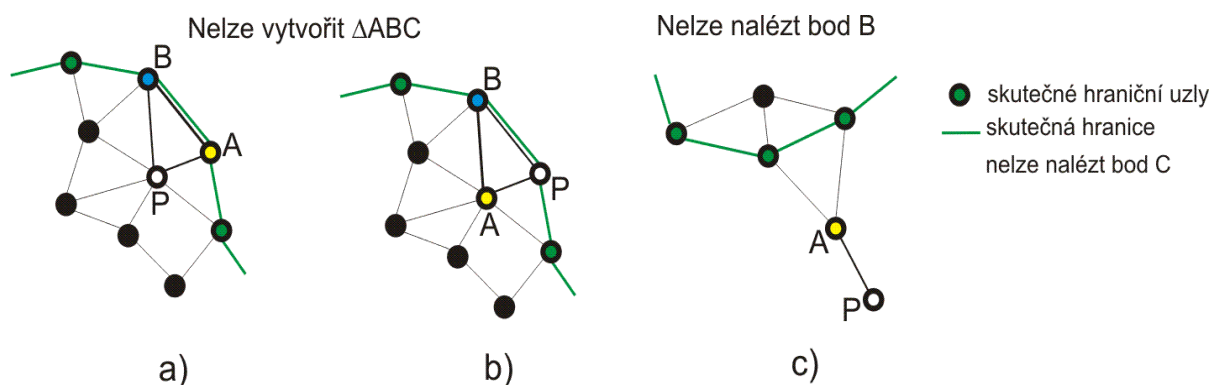


Obr. 6. Chyby algoritmu CGN – na stejné přímce.

- 2) Druhá nedokonalost návrhu nastane, pokud určitý uzlu P nenalezne jakýkoliv C. Vhodné body B i A sice nalezeny byly, ovšem bez třetího uzlu nelze vytvořit trojúhelník. Algoritmus CGN nevyhodnocuje (vnitřní, vnější) uzly, ke kterým neexistuje jediný trojúhelník ($\triangle ABC$), a tímto nejsou určeny jako vnější (viz Obr. 7. a)). Pozn. v grafu (Obr. 17, Obr. 18., Obr. 19.) jsou barevně zaznačeny pouze uzly vyhodnoceny jako vnější (červeně určeny chybně a zeleně správně).

Může ovšem nastat případ znázorněn na Obr. 7. b) a tedy tím, že nebyl nalezen jediný trojúhelník, je uzel P nevyhodnocen. → Uzel je tedy vyhodnocen nesprávně. Tyto situace nastávají častěji u řídkých sítí.

- 3) Obdobná situace jako v předchozím případě nastane na Obr. 7. c). V tomto případě pro uzel P existuje jediný sousední uzel (A), a tedy nelze nalézt další body B, C. Viz předchozí případ, uzel P je nevyhodnocen a je vyhodnocen opět nesprávně. Stojí za uvažovou, zda nebudou-li nalezeny všechny body (A, B, C) uzel vyhodnotit jako vnější či vnitřní, a také možné množství těchto kritických případů.



Obr. 7. Chyby algoritmu CGN – nenalezeny body B, C.

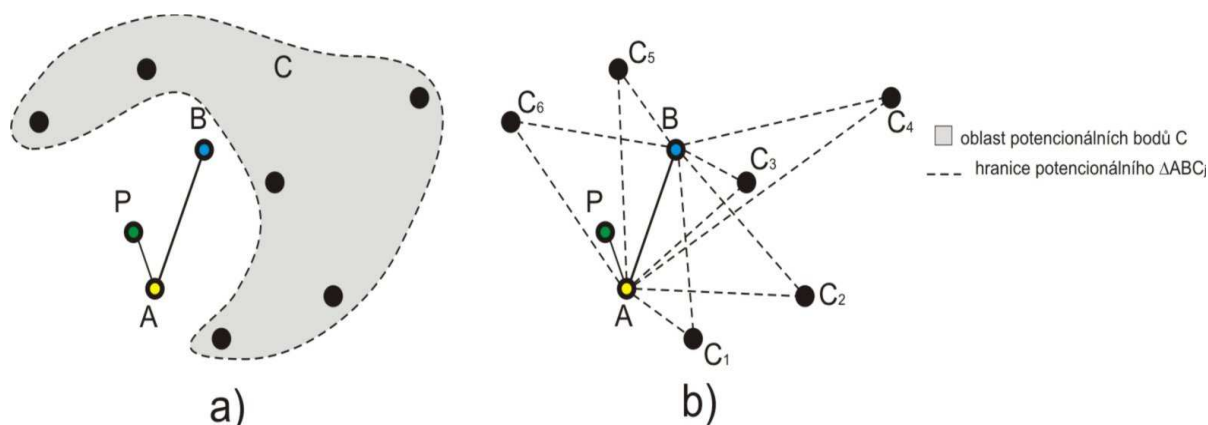
- 4) Další nedokonalost navrženého algoritmu je znázorněna na Obr. 8. a). Týká se situací, kdy jsou nalezeny všechny body (A, B, C) trojúhelníku, ale i přesto je bod P vyhodnocen chybně. Vzhledem k $\triangle ABC$ (vyznačeno plnou čarou) bod P leží vně, ale v síti je vnitřním uzlem. Tento fakt předkládá myšlenku, zda jsou body A, B, C vhodně vybrány a při jiném výběru (A_2) by daný bod P byl vyhodnocen stejně. Výběr vhodných bodů může tedy značně ovlivnit výsledný trojúhelník předkládaný k rozhodovací podmínce. Bod P vybírá jako první bod (A) z uzlů číslo 1, 2, 3, 4 a podle podmínky v Tab. 2. bodě 1. je vybrán nejbližší uzel tedy č. 1. Výběrem prvního uzlu (A) se mohou podstatně změnit možnosti volby dalších uzlů (B, C) a také výsledky (vnější, vnitřní). Například, kdyby jako počáteční bod A byl zvolen uzel č. 3 a k němu vytvořen trojúhelník (vyznačeno tečkovaně) dle již neměnných pravidel CGN, pak bod P by se nacházel uvnitř $\triangle ABC$ a byl vyhodnocen jako vnitřní. Pozn. body A a A_2 nejsou ve vzájemném rádiovém dosahu, tedy se nejedná o situaci se dvěma trojúhelníky $\triangle ABC$ a $\triangle ABD$ (na obrázku $\triangle A_2BC$)
- 5) Obdobným problémem jako bod 4) se zabývá i tento bod. Je-li první bod A vybrán jako nejbližší k uzlu P, pak existuje několik možných potencionálních uzlů B. CGN vybírá pouze jeden uzel nejvíce vzdálený od bodu A (viz Tab. 2. bod 3.), tedy bod B_1 . Mohl by nastat obdobný případ jako na Obr. 8. a), kde výběrem jiných uzlů lze ovlivnit správnost výsledků. Tato situace je naznačena na Obr. 8. b) stejně jako rozdílné řešení při výběru B_1 (vyznačeno plnou čarou) či B_2 (vyznačeno tečkovaně). Při výběru bodu B_1 bude uzel P vyhodnocen jako vnější a tudíž by se jednalo o chybné vyhodnocení. V případě výběru bodu B_2 bude uzel P ležet uvnitř sítě i trojúhelníku $\triangle AB_2C_2$. Rozdílné výsledky platí i pro různé uzly C (C_1, C_2, \dots) a touto myšlenkou je inspirován algoritmus BRC (uveden níže). Uzly A, B jsou vybrány dle podmínek Tab. 4., body 3, 6 a všechny možné potencionální body C jsou testovány. Oba algoritmy CGN a BRC mají vysoké procento úspěšnosti, tedy ve většině

Výše bylo popsáno 5. kritických situací, které algoritmus CGN vyhodnotí mylně, a tyto poznatky by měly být použity k stanovení základních pravidel při tvorbě dalšího algoritmu.

Další ze známých algoritmů pro detekci hraničních uzlů je algoritmus BRC (Boundary Recognition using Cset), který byl inspirován myšlenkou vytvářet vhodné trojúhelníky ke každému uzlu P, jak je tomu u CGN. Hlavní modifikace tkví v dokonalejším výběru uzlů A, B, C (nehledá se uzel D), a také v testování všech možných variant trojúhelníků ABC souboru $C = \{C_1, C_2, \dots, C_i\}$. [4]

3.2.1 Návrhový algoritmus

Algoritmus BRC se skládá ze dvou základních částí: výběr souboru C (createCset) a rozpoznání hranic (boundaryRecognition). V první části viz pseudokód Tab. 4. jsou vybrány každému uzlu P vhodné sousední body A , B a soubor C . Nejprve se pro bod P vybere z tabulky sousedních uzlů ($N(P)$) bod A s nejmenší vzdáleností. Poté se hledá bod B tak, aby rozdíl mezi vzdálenostmi d_{PA} a d_{PB} byl minimální, a zároveň vzdálenost bodu B od A byla co největší viz Obr. 9 a). Je použita se rozhodovací podmínka $B = \arg [\max (d_{AB} - d_{PA} - d_{PB})]$, a také musí platit podmínka, že bod B je v tabulce sousedních uzlů bodu P i A tzn. $B \in N(P) \& N(A)$. Cílem tohoto postupu je vhodným způsobem vybrat uzly A a B , jež vytvoří jakousi základnu pro $\triangle ABC$ s bodem P co nejvíce uprostřed. Dále je zde v tečkované oblasti soubor všech možných bodů C . Všechny uzly (n), které splňují podmínku $N(P) \& N(A) \& N(B)$ naleží do souboru uzlů C a poslouží k posouzení, zda P je hraniční či vnitřní uzel. Obr. 9. b) ilustruje všechny možné body C a k nim vytvořeny $\triangle ABC_1, \triangle ABC_2, \dots, \triangle ABC_6$. Je zřejmé, že nastal 5x případ, kdy P je hraniční uzel a 1x vnitřní. Nastane-li alespoň jeden případ, kdy P je uvnitř trojúhelníku (v našem případě $\triangle ABC_6$) situace bude vyhodnocena tak, že P je hraniční. Může ovšem nastat jeden z extrémních případů, kdy nelze nalézt žádný uzel C , a tím sestrojít $\triangle ABC$. Obdobná situace nastane i v případech, že nelze nalézt ani bod B . V těchto situacích s velkou pravděpodobností uzel P leží na okraji pole, a z tohoto důvodu nelze nalézt ostatní body. P je proto vyhodnocen algoritmem BRC jako hraniční uzel. Viz Tab. 4. body 5,8. [4]



Obr. 9. Kroky pro výběr A , B , C algoritmu BRC.

createCset (n_i)	
Vstupy:	Uzly P
Výstupy:	Uzly A, B, C, Hranicni
1.	Hranicni = nepravda
2.	A, B, C = 0
3.	for $\forall n \in N(P)$ do $\#N(P)$...tabulka sousedních uzlů n_i výběr bodu A s $\min(d_{PA})$ end
4.	vytvořit $N(PA) = N(P) \cap N(A)$
5.	if $N(PA)$ je prázdná return Hranicni = pravda \rightarrow zastavit
6.	for $\forall n \in N(PA)$ do výběr bodu B s $\max(d_{AB})$ & $\min(d_{PA} - d_{PB})$ end
7.	vytvořit $N(PAB) = N(PA) \cap N(B)$
8.	if $N(PAB)$ je prázdná return Hranicni = pravda \rightarrow zastavit else $C = N(PAB)$ end
9.	return A, B, C, Hranicni [4]

Tab. 4. Pseudokód algoritmu BRC část createCset.

Část nazvaná jako výběr souboru C (createCset) posloužila k určení vhodných sousedních uzlů pro body A, B, C_i a vytyčení extrémních případů, kdy nelze nalézt všechny body pro trojúhelník ABC. Zbývá posoudit dle rozhodovací podmínky, zda se jedná o hraniční uzly. K tomuto poslouží druhá část algoritmu BRC nazvaná rozpoznání hranic

(boundaryRecognition). Rozhodovací podmínka bude stejná jako u IP aby bylo možné porovnat výsledky.

Druhá část algoritmu rozpoznání hranic pracuje s hodnotami (A, B, C a Hranicni) z části Výběr souboru C a dále je bude analyzovat viz Tab. 5. Do smyčky v Tab. 5 bod 3. vstupují pouze případy, kdy se nejedná o hraniční body tzn. Hranicni = nepravda. Pro každý vybraný bod C_1, C_2, \dots, C_j jsou vypočteny dílčí obsahy $S_{P,A,C}$, $S_{P,B,C}$ a $S_{P,A,B}$ a jeden obsah $S_{A,B,C}$ trojúhelníku $\triangle ABC_j$. V bodě 3. na 4 řádku je hlavní rozhodovací podmínka $S_{A,B,C} = S_{P,A,C} + S_{P,B,C} + S_{P,A,B}$ (stejná jako u IP) a v případě, že vyhoví, znamená to, že uzel P je vnitřní (Hranicni = nepravda) viz také ilustrace Obr. 5. d). Druhá možná situace, která může nastat, je že P bude ležet mimo $\triangle ABC$ a tudíž by platilo, že součet ploch dílčích trojúhelníků (PAC, PBC, PAB) je větší než plocha $S_{A,B,C}$. [4]

boundaryRecognition (P)	
Vstupy:	Uzly A, B, C, Hranicni
Výstupy:	Hranicni
1.	Hranicni = pravda
2.	[A, B, C, Hranicni] = createCset (P)
3.	if (Hranicni == nepravda) for $\forall C_j \in C$ do vypočet $S_{A,B,C}, S_{P,A,C}, S_{P,B,C}, S_{P,A,B}$ if $S_{A,B,C} = S_{P,A,C} + S_{P,B,C} + S_{P,A,B}$ Hranicni = nepravda \rightarrow zastavit end end end
4.	return Hranicni [4]

Tab. 5. Pseudokód algoritmu BRC část boundaryRecognition.

3.3 Algoritmus BRB (Boundary Recognition using Bset)

V této části bude popsán nový algoritmus výběru krajních uzlů v náhodně rozmístěných sítích. Hlavní myšlenkou je vyvarovat se chybným detekcím v krizových situacích, a tím minimalizovat chybovost detekce. Každému bodu P bude vybrán jeden bod A a několik potencionálních bodů B, C .

3.3.1 Návrhový algoritmus

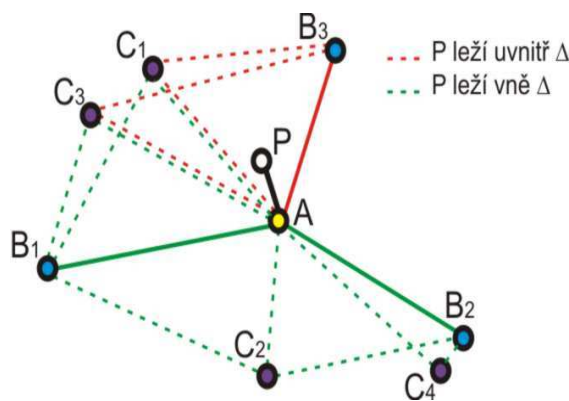
Algoritmus BRB se skládá z části výběr souboru B (createBset) a rozpoznání hraničních uzlů (boundaryRecognition). Dle pravidel pseudokódu createBset viz Tab. 6. Je každému uzlu P vybrán jeden uzel A , tedy uzel s nejmenší vzdáleností. Jako druhý uzel (B) se vyberou všechny potencionální uzly ležící v $N(P)$ a $N(A)$. Nastane-li případ viz Obr. 7. c), kdy nelze nalézt jediný uzel B , tak se záznam o uzlu P ignoruje a další bod (3) tabulky se neprovede celý (tzn. nevytvářet záznam o uzlu P a „ P je okrajový bod sítě“). Tuto situaci řeší podmínka bodu 3. v Tab. 7. Jsou-li nalezeny tři uzly B , pak ke každému z nich jsou přiřazeny všechny potencionální uzly C za podmínky $n \in N(P) \& N(A) \& N(B)$. Záznam pro uzel P pak může obsahovat např. 6 řádků ($B_1 \rightarrow C_1, C_2, C_3$ pro $B_2 \rightarrow C_2, C_4$ a pro $B_3 \rightarrow C_1, C_3$) viz Obr. 10. Vytvářením většího množství trojúhelníků algoritmus dospěje k menšímu množství chybně vyznačených hraničních uzlů. Tedy neměly by nastat případy, kdy výběrem jiného uzlu B dojde k rozdílnému výsledku viz Obr. 8 b). Obdobná situace platí i pro výběr bodu C . Část algoritmu createBset slouží pouze pro výběr všech možných trojúhelníků a na něj navazuje boundaryRecognition.

createBset (n_i)	
Vstupy:	Uzly P
Výstupy:	Uzly A, B, C, P je okrajový bod sítě
1.	for $\forall n \in N(P)$ do Výběr bodu A s $\min(d_{PA})$ end
2.	for $\forall n \in N(P) \& N(A)$ do

	Výběr všech potencionálních bodů B_i if $N(P) \cap N(A)$ je prázdná nevytvářet záznam o uzlu P end end
3.	for $\forall n \in N(P) \& N(A) \& N(B)$ do Výběr všech potencionálních bodů C_j if $N(P) \cap N(A) \cap N(B)$ je prázdná nevytvářet záznam o uzlu P a „P je okrajový bod sítě“ end end
4.	Pokračování P, A, B, C, P je okrajový bod sítě

Tab. 6. Pseudokód algoritmu BRB část createBset (n_i).

Na část algoritmu createBset (popsán výše) navazuje algoritmus boundaryRecognition pro vyhodnocení nalezených trojúhelníků $\Delta AB_i C_j$ včetně určitých krizových situací. Pseudokód algoritmu boundaryRecognition je popsán v Tab. 7., a poslouží za prvé k testování jednotlivých nalezených trojúhelníků, a za druhé ke konečnému verdiktu bod P je vnitřní/vnější vzhledem k celé síti. Do smyčky (for) v bodě 2. vstupují pouze případy, kdy byly nalezeny A, B, C a tedy pro bod P existuje alespoň jeden záznam. Uvnitř smyčky jsou vypočteny plochy dílčích trojúhelníků ΔPAC , ΔPBC , ΔPAB a celková plocha ΔABC dle Heronova vzorce. Dále jsou tyto výpočty podrobeny rozhodovací podmínce $S_{A, B, C} = S_{P, A, C} + S_{P, B, C} + S_{P, A, B}$ a vyhoví-li uzel P, je uvnitř $\Delta AB_i C_j$. V opačném případě je uzel P vyhodnocen vně $\Delta AB_i C_j$. Nastane-li případ (jak bylo uvedeno výše), kdy bod P obsahuje 6. Záznamů, pak každý z nich bude vyhodnocen jako uvnitř/vně k $\Delta AB_i C_j$. Tedy 6 x vně a 2 x uvnitř viz Obr. 10.



Obr. 10. Rozdílné výsledky pro různé body B.

V bodě (3.) je konečná rozhodovací podmínka. Pokud počet případů vyhodnocených jako P je uvnitř ΔAB_iC_j je roven 0, pak se jedná o okrajový bod. Jako okrajové body jsou vybrány také případy na Obr. 7. b) a c) tedy s nenalezenými vhodnými body B, C . Fakt, že počet nalezených případů, kde P leží vně, je větší než počet uvnitř, je nepodstatný, protože existuje-li alespoň jeden trojúhelník s P uvnitř, pak se nemůže jednat o okrajový uzel. Algoritmus předpokládá, že situace naznačena na Obr. 7. a) nastane méněkrát než případ Obr. 7. b). V opačném případě by bylo potřeba pozměnit pravidla např., je-li nalezen alespoň jeden bod A a B , pak bod P je s velkou pravděpodobností uvnitř sítě. Tímto přístupem by algoritmus BRB chybně detekoval situace, kdy P leží na okraji sítě a v tabulce sousedních uzlů jsou pouze dva uzly a také Obr. 7. b). Tzn., jsou nalezeny pouze body A a B . Proto situace naznačená na Obr. 7. a) není vyřešena a předpokládá se, že nastane co nejméně.

boundaryRecognition (P)	
Vstupy:	P, A, B_i, C_j, P je okrajový bod sítě
Výstupy:	P je okrajový bod sítě
1.	$[A, B, C, P \text{ je okrajový bod sítě}] = \text{createBset}(P)$
2.	<p>for $\forall B_i \in B, C_j \in C$ do</p> <p>Vypočet $S_{A,B,C}, S_{P,A,C}, S_{P,B,C}, S_{P,A,B}$ pomocí Heronova vzorce</p> $S_{A,B,C} = \sqrt{[s * (s - a) * (s - b) * (s - c)]}$ <p>,kde $s = (a + b + c)/2$ a a, b, c jsou délky tří stran trojúhelníka</p> <p>if $S_{A,B,C} = S_{P,A,C} + S_{P,B,C} + S_{P,A,B}$</p> <p>„$P$ je uvnitř ΔAB_iC_j“</p>

	<pre> else „P je vně ΔABC“ end end </pre>
3.	<pre> if počet (P je uvnitř ΔABC)==0 „P je okrajový bod sítě“ end </pre>

Tab. 7. Pseudokód algoritmu BRB část boundaryRecognition (P).

3.3.2 Shrnutí BRB

Algoritmus BRB se snaží eliminovat velké množství chybně vyznačených vnějších uzlů a také se vyvarovat některým kritickým situacím. Tedy především chybám algoritmu CGN (viz 3.1.5 Chyby algoritmu CGN) a docílit co nejmenší chybovosti detekce. Lze tvrdit, že algoritmus BRB je odolný vůči krizovým situacím na Obr. 7. b), c), Obr. 8. b) a Obr. 6. b). Jelikož nelze jednoznačně rozhodnout situaci na Obr. 6. a) a b), byla zvolena strategie, pokud P leží na hraně ΔABC , pak je vnitřním uzlem. Poslední nevyřešena krizová situace je na Obr. 8. a). Tedy jiným výběrem prvního uzlu (A) lze ovlivnit výsledek. Řešením by bylo vybírat všechny možné body A, k nim všechny možné B a taktéž všechny body C. Toto řešení by bylo samozřejmě výpočetně více náročné. O výpočetní náročnosti blíže pojednává kapitola 2.3 Složitost algoritmu. Také komunikační náročnost bude u algoritmu BRB větší než u CGN a BRC.

3.4 Algoritmus OBR (On boundary recognition without location information)

3.4.1 Úvod

Stejně jako u algoritmů CGN, BRC i BRB, tak i v případě OBR jednotlivé uzly nejsou vybaveny systémem GPS, a proto je zapotřebí určit hraniční a vnitřní uzly na základě vzájemných vzdáleností. Algoritmus OBR definuje hranice pomocí nejkratších cyklů. Problém vymezení hranic je popsán níže. Přístup je schopen odstranit předpoklad o hustotě uzlů, a tím

poskytuje dobré řešení pro řídké i husté sítě. Je potřeba si představit geometrickou konstrukci tzv. schémata, která zaručují, že jednotlivé uzly leží uvnitř schématu pro všechna nasazení senzorové sítě.

3.4.2 Předpoklady

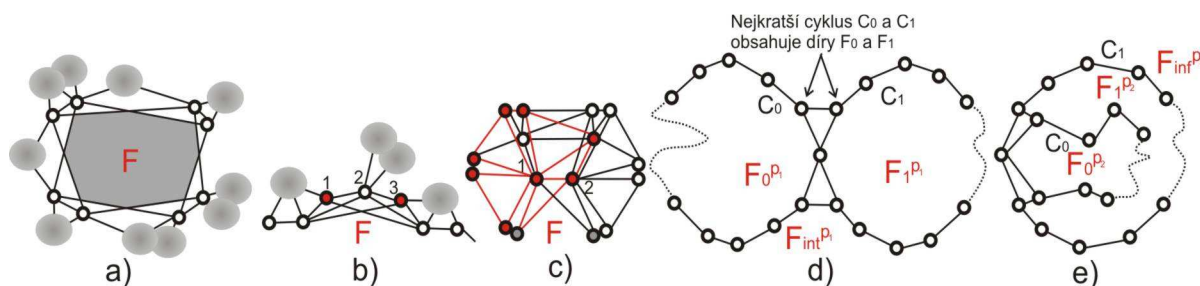
Existují tři základní předpoklady:

1. Předpoklad určitého rozmístění senzorů v prostoru. Nesmí se jednat o síť s vnitřní hranicí. Viz Obr. 2. a).
2. Nejkratší vzdálenost mezi dvěma uzly poskytuje přiměřenou geometrickou hustotu. Tzn. je potřeba mít dostatečně hustou síť.
3. Algoritmus hledá několik typů modelů tzv. „květy“, které pak budou dále rozšířeny či slučovány do složitějších fází algoritmu. Podrobněji vysvětleno viz níže.

Tento přístup volí přístup především třetí skupiny. Jednoduchost a obecnost vzorů výrazně snižuje požadavky na hustotu uzlů v síti a poznání schématu. Úroveň vnoření je definována jako skupina uzlů se stejnými vzdálenostmi. [3]

3.4.3 Hranice senzorové sítě

Hranice sítě je komplexní prostorová vlastnost ohraničená přímými liniemi ve známém 2D prostoru. Síť může mít několik hranic: vnější hranice (nekonečná oblast) viz Obr. 2. b) a vnitřní hranice (konečná oblast) viz Obr. 2. a). Obvod hranice je jedinečný a kontinuální pro dané rozložení. Za povšimnutí stojí, že ve vnitřní oblasti se nemusí nacházet uzly na hranicích viz Obr. 11. a). Obdoba platí pro vnější oblast, kde uzly 1 a 3 nemusí nutně tvořit spojený cyklus viz Obr. 11. b). Tyto uzly by mohly patřit k nejkratšímu cyklu to je ovšem v rozporu s myšlenkou „Existuje pouze jeden cyklus pro vnější obvod“. [3]

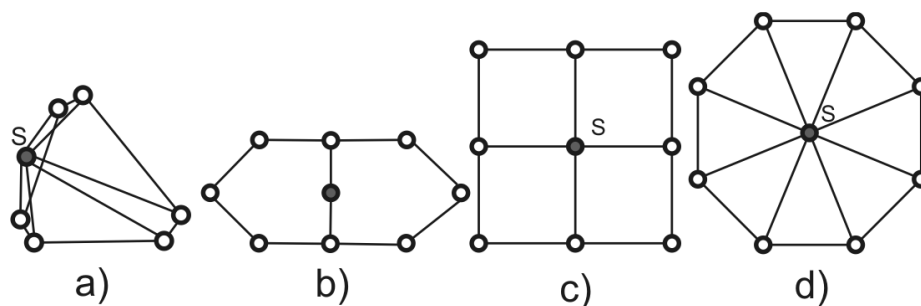


Obr. 11. Problémy související s definicí hranic.

Příklady, kdy uzel 1 leží uvnitř „květu“ (znázorněno červeně) je uveden na Obr. 11. c). Uzel s šedou barvou znázorňuje přerušení kontinuity hranice. Je zapotřebí zvolit minimální velikost pole. Na Obr. 11. d) je znázorněna situace, kde nejkratší cyklus C_0 a C_1 obsahuje vnitřní oblast F_0 a F_1 . Předpokládá se, že $|C_1| < |C_0|$, a pokud je uznána vnitřní oblast o velikosti nejméně \min_C s $|C_1| < \min_C < |C_0|$, pak by měly být detekovány algoritmem OBR oblasti F_0 a F_1 . Na Obr. 11. e) jsou znázorněny dvě oblasti (F_0^{p2} a F_1^{p2}) uvnitř oblasti F_{inf}^{p2} příliš malé na to, aby byly kvalifikovány jako konečná oblast (ohraňována vnitřní hranicí C_1). Důvodem je, že hranice C_0 je menší než \min_C . [3]

3.4.4 Vzory

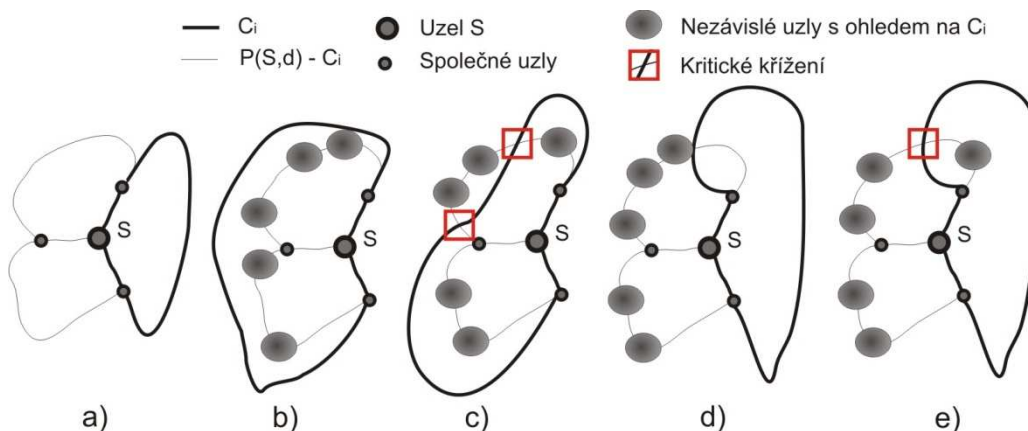
Je zapotřebí zavést pojem subgraf, který garantuje, že určité uzly budou ležet uvnitř tzv. jádra vzoru. Na Obr. 12. a) je znázorněn případ, kdy se nejedná o subgraf. Složitější subgrafy nazvány jako vzory jsou na Obr. 12. b), c), d). Jako nejmenší možný model je pak definována konstrukce na Obr. 12. d). Každý vnitřní uzel S potřebuje znalosti o komunikačních vazbách mezi sousedními uzly. [3]



Obr. 12. a) Nevyhovující konstrukce pro vzor; b) a c) Vzory; d) Model.

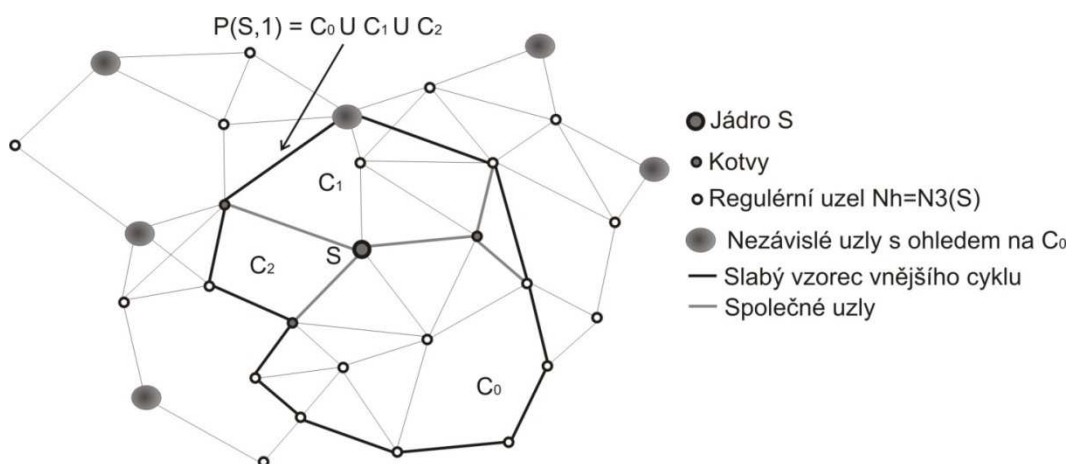
Jako další termín je zapotřebí definovat chordless cyklus C jako polynom, který rozkládá plochu na nekonečnou oblast a alespoň jednu konečnou oblast (nejedná se o vnější hranici a pro rozsáhlost podrobnější definice chordless cyklu viz originální text ze zdroje [3]). Zda lze umístit subgraf uvnitř chordless cyklu rozhodují podrobné podmínky v textu [3]. Každý chordless cyklus je omezen maximální oblasti závislé na délce chordless cyklu. Na Obr. 13. jsou znázorněny tři možné kombinace jednoho chordless cyklu s ostatními: může je obsahovat b), může je protínat c), d), e), a nebo může ležet na jiné straně a), d), e). Chordless cyklus C_i je nazván jako odražený viz případy b), c). Pokud S leží mimo konstrukci, pak buď jeden cyklus obsahuje všechny ostatní, nebo se jedná o kritický průsečík nejméně dvou

chordless cyklů. Příklad na Obr. 13. b) je zamítnut (není vhodný pro vzor), protože jeden cyklus obsahuje všechny ostatní. Příklad c) a e) jsou zamítnuty z důvodu kritických průsečíků. Průsečík v případě d) není kritický, a obě situace a) a d) jsou povolené vzory. [3]



Obr. 13. Kombinace tří chordless cyklů.

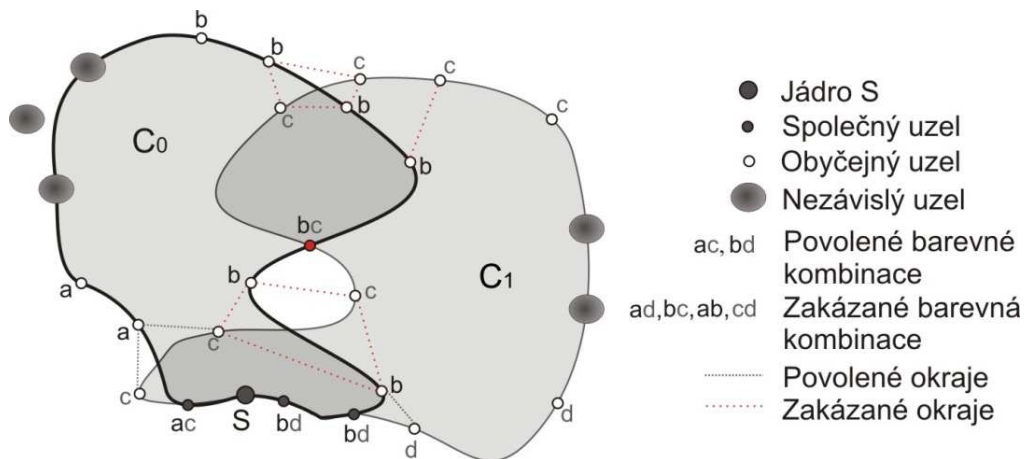
Slabý vzor je definován jako několik chordless cyklů (C_i) a formálně: $P(S, 1) = C_0 \cup \dots \cup C_{n-1}, 2 \leq n \leq 5$. Mohutnost vzoru je závislý na počtu složených vzorů a každý vzor obsahuje jednu kotvu. Skládá-li se slabý vzor s nejméně třemi chordless cyklů (C_i), pak se každý pár $C_i, C_{(i+1) \bmod n}$ dělí o jednu kotvu viz Obr. 14. Pokud je slabý vzor složen z přesně dvou chordless cyklů pak je kotva sdílena oběma cykly. [3]



Obr. 14. Rozšíření nezávislého souboru vlastností.

Leží-li S na okraji konstrukce, pak vložením nejméně dvou chordless cyklů se musí tyto cykly protínat. Ovšem také je zapotřebí, aby konstrukce obsahovala alespoň jeden nezávislý uzel s ohledem ke zbývajícím. Příklad těchto konstrukcí je uveden na Obr. 15., kde se rozlišuje

vrchol základny (červený bod bc) viz a) a okrajový vrchol kritického průsečíku viz b). Protože je zapotřebí zjistit kritické průsečíky, je požadavek, aby cykly sdílely alespoň jeden uzel (červený uzel bc) viz a) nebo jeden z červeně vyznačeného okraje (b, c, b, c) viz b). Výše bylo zmíněno, že jeden chordless cyklus nemůže obsahovat další chordless cykly viz Obr. 13. b), c), e). Nicméně jsou nalezeny konflikty (průsečíky mezi C_0 a C_1), a ty musí mít povolenou barevnou kombinaci. Proto nejsou nezávislé uzly umístěny mezi jádro S a povolené průsečíky. [3]



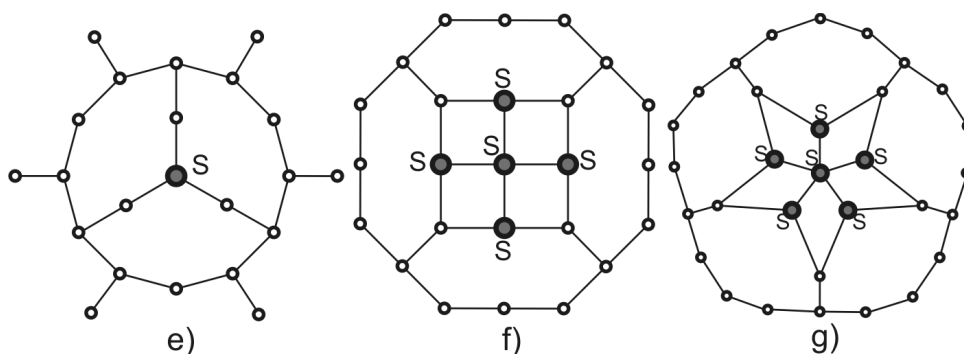
Obr. 15. Test zbarvení a typy kritických průsečíků.

Vzory uvedeny na Obr. 12. jsou již nedostačující, a proto je rozšíření na Obr. 16. Příklady b) a c) jsou složitější varianty a řešení zaručeně vnitřních uzlů se bude provádět obecnými přístupy. Silný vzor je definován jako několik chordless cyklů (C_i) formálně: $P^*(S, d) = C_0 \cup \dots \cup C_{n-1}, 2 \leq n \leq 5, d \geq \frac{\sqrt{2}}{2}$ Samozřejmostí je dodržení podmínek slabých vzorů.

Z vlastností a definic slabých a silných vzorů lze často odvodit další prostorové informace sensorové sítě, a tím lze optimalizovat vzor algoritmu rozpoznání. Např.

Garance vzdálenosti: Silné i slabé vzory garantují, že jádro leží uvnitř vzoru. Silný vzor pak dodatečně garantuje vzdálenost pro jednotlivá jádra.

Solidnost: Přístup ke vzorům, na něž se vztahuje garance, že jádro odpovídajícího uzlu leží uvnitř sítě pro všechny rozšířené vzory s $d \geq \frac{\sqrt{2}}{2}$. [3]



Obr. 16. Rozšířené vzory.

3.4.5 Hranice rozpoznávání algoritmu

Algoritmus pro rozpoznání hranic je prezentován stanovenými matematickými základy a nezbytnými podmínkami pro vzory. Cílem algoritmu rozpoznání hranic je najít slabé a silné vzory v síti. Je-li garantováno, že uzel leží uvnitř sítě, je uzlu přiřazena úroveň 1. Pokud je navíc zaručeno, že musí mít alespoň určitou vzdálenost od kraje, pak je uzlu přiřazena úroveň 2. U všech ostatních uzlů se předpokládá, že jsou hraniční a získaly úroveň 0. Maximální i minimální délka chordless cyklu je limitována. Po dokončení rozpoznávacího hraničního algoritmu se algoritmus spustí znova, ale používá pouze uzly, které byly identifikovány jako jádra silných vzorů. Algoritmus se provádí opakovaně pokaždé, když se změní graf a zvýší se úroveň (pouze pokud jsou nalezeny silné vzory). [3]

Algoritmus OBR je pro svou rozsáhlost popsán velice zkráceně a podrobněji viz [3]. Z důvodu velké náročnosti a ne realizace algoritmu OBR byly vybrány k simulacím algoritmy CGN, BRC a BRB.

4. Analýza algoritmů pro detekci hraničních uzlů

V následujícím textu bude přibližně popsán postup simulačního programu, blíže rozebrány některé úskalí a nedokonalosti algoritmů a porovnání dosažených výsledků algoritmů CGN, BRC a BRB.

4.1 Stručný popis simulačního programu

Algoritmus pro výběr hraničních uzlů (BRB) je naprogramována a simulována v prostředí Matlab verze 7.10.0.499 (R2010a). Hlavní program je rozdělen na dvě části tzn. dva M-file (funkce, skript v souboru) „spustBRB.m“ a „BRB.m“, kde „BRB.m“ představuje samotný algoritmus BRB a soubor „spustBRB.m“ slouží pro spouštění jednotlivých simulací s různými vstupními parametry (databáze uzlů, počet uzlů, rádiový dosah).

Nejprve bude popsán soubor „spustBRB.m“, který načte databázi vstupních uzlů ze souboru „networksDB.mat“ (pozn. algoritmy CGN, BRC i BRB používají stejnou vstupní databázi). V databázi je 3 x 10 různých modelů sítí s hustotou 8 – 26 uzlů. Rádiový dosah je volen pro všechny modely 25 metrů. S rostoucím číslem modelu dochází k zhušťování sítě. Tímto postupem bylo dosaženo, že s každým dalším modelem jsou jednotlivé uzly blíže sebe, a tím pádem mají ve svém konstantním rádiovém dosahu větší množství sousedních uzlů. Dalším vstupním parametrem pro vyhodnocení jednotlivých simulací je seznam správných hranic, jenž je načten ze souboru „databaseBoundaryIDs.mat“. Výstupní parametry skriptu „BRB.m“ jsou ukládány do „finishtabulka“ a z ní je pak vytvořen graf boxplot (vysvětlen níže viz kap. 4.3 Úspěšnost a chybovost detekce) pro úspěšně a chybně nalezené hraniční uzly. Z výstupních parametrů jsou sestaveny také grafy průběhů úspěšnosti pro jednotlivé algoritmy v závislosti na zvětšujícím se stupni uzlů. Jelikož jsou křivky tvořeny pouze 10-ti hodnotami byly jednotlivé křivky vhodně vyhlazeny.

Dalším souborem je „BRB.m“, kde tělo programu tvoří tyto hlavní funkce: „printNet, BodA, BodyB, BodyC, boundaryRecognition a printNet2“. Jednotlivé funkce budou vysvětleny jen okrajově. Funkce printNet sestaví tabulku sousedních uzlů pro jednotlivé uzly a vypočte vzájemnou vzdálenost na základě vstupní databáze a rádiového dosahu. Jak bylo řečeno výše, čím vyšší číslo modelu, tím je rozložení sítě hustší a počet sousedních uzlů bude větší. Funkce „BodA“ najde pro každý bod P nejvhodnější bod A podle kritérií v Tab. 6.

a všechny možné body B a C viz „BodB, BodC“. Další funkce Ippabcd vypočte obsahy jednotlivých trojúhelníků (pouze pokud byl nalezen vhodný C) ΔABC , ΔPAC , ΔPBC , ΔPAB a rozhodne, zda bod P leží uvnitř nebo vně ΔABC . Směrodatné je pouze pokud bod P leží uvnitř ΔABC . V poslední hlavní funkci „printNet2“ se budou porovnávat vypočtené vnější uzly se skutečnými hraničními. Pro lepší názornost funkce dále vykreslí graf rozložení uzlů a vypočte parametry úspěšnost a chybovost detekce nalezených hraničních uzlů viz „finishtabulka“.

Koncepce programu pro výpočet CGN se nějak podstatně neliší. Největší změny jsou ve výběru pouze jednoho bodu B, C a D. Funkce „boundaryRecognition“ vypočte obsahy jednotlivých trojúhelníků (pouze pokud byl nalezen vhodný C) ΔABC , ΔPAC , ΔPBC , ΔPAB a rozhodne, zda bod P leží uvnitř nebo vně ΔABC , popřípadě pokud existuje ΔABD , provede obdobné výpočty.

Pro spuštění celkové simulace, tedy 10-ti různých modelů sítí s 50, 100 a 400 uzly je zapotřebí vybrat v editoru v menu Debug → Run spousteci.m nebo zmáchnout klávesu F5. Po dokončení výpočtu každé jednotlivé sítě program vykreslí pro názornost graf rozložení hraničních a vnitřních senzorů viz Obr. 17., Obr. 18. a Obr. 19.

Parametry simulace jsou:

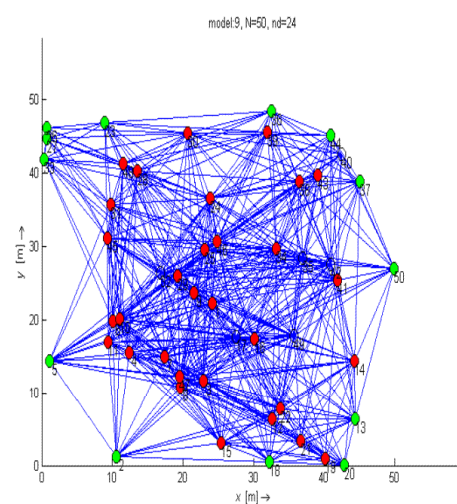
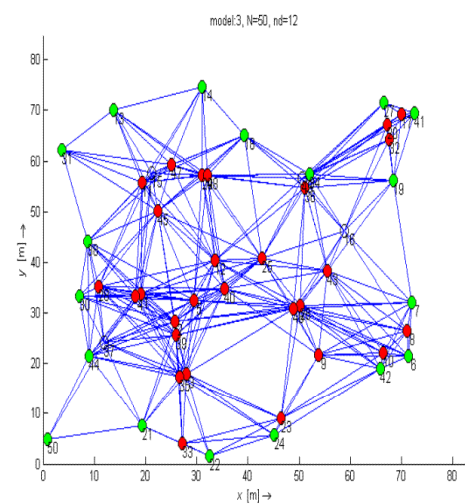
Parametry simulace	Hodnoty
Databáze vstupních uzlů	networksDB.mat (3x10 modelů)
Počet uzlů	$numNodes = 50, 100, 400$
Rádiový dosah uzlů	$R = 25$ m
Stupeň uzlů	$nd = 8, 10, 12, 14, 16, 18, 20, 22, 24, 26$
Chyba určení vzdálenosti	$\varepsilon_r = 0$ % (zatím se uvažuje ideální prostředí)

Tab. 8. Parametry simulace algoritmu CGN, BRC a BRB, ideální prostředí.

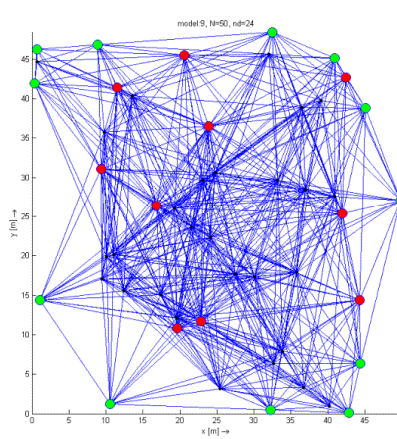
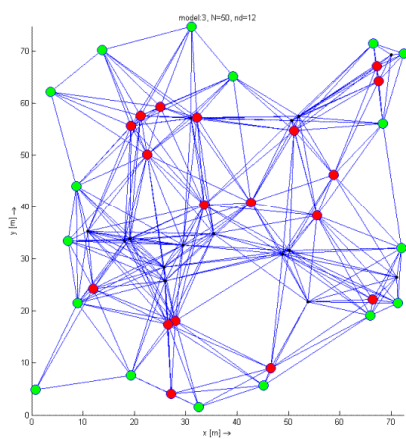
4.2 Rozbor grafů

Z jednotlivých sítí pro 50, 100 a 400 uzlů byly vybrány k porovnání modely č. 3. a 9., kde model 9 má větší hustotu než model 3. Zbylé grafy obsahuje příložený nosič DVD. Budou porovnány algoritmy nazvané CGN (horní dva grafy), BRC (prostřední dva grafy) a BRB (dolní dva grafy). Na Obr. 17. jsou znázorněny sítě s 50 uzly, na Obr. 18. jsou sítě se 100 uzly a největší simulované sítě se 400 uzly viz Obr. 19. V jednotlivých grafech jsou zeleně vyznačeny úspěšně nalezené skutečné hraniční uzly a červeně jsou naznačeny chybně vybrané uzly.

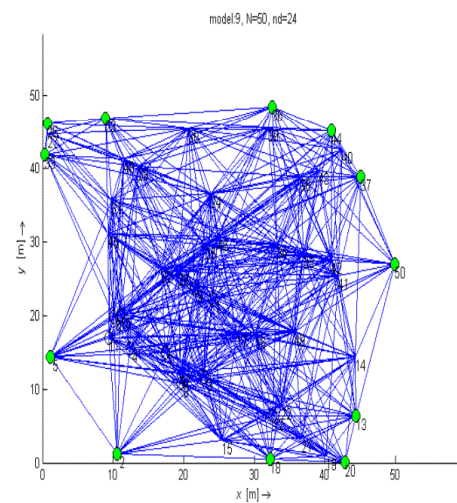
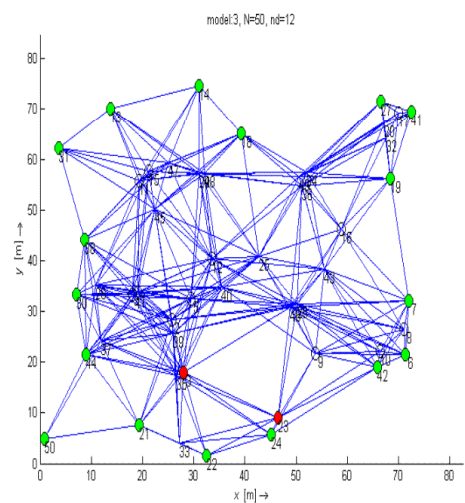
CGN



BRC

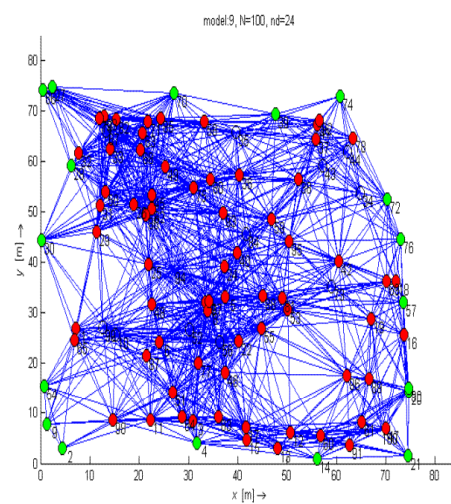
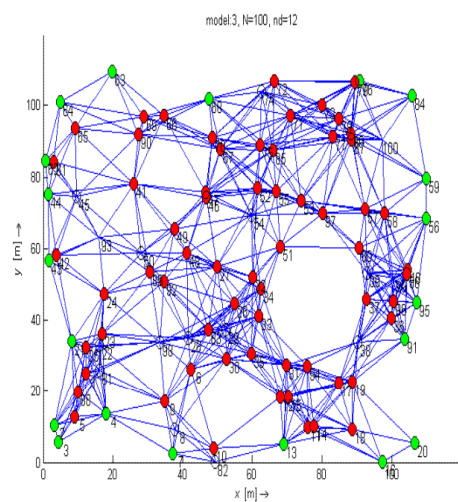


BRB

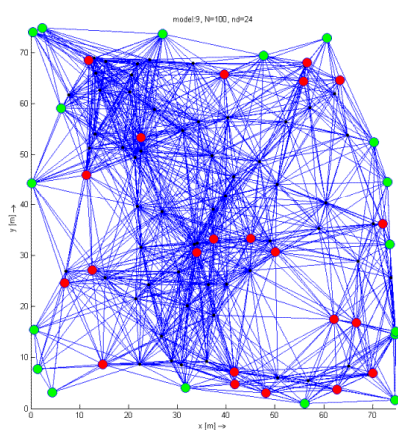
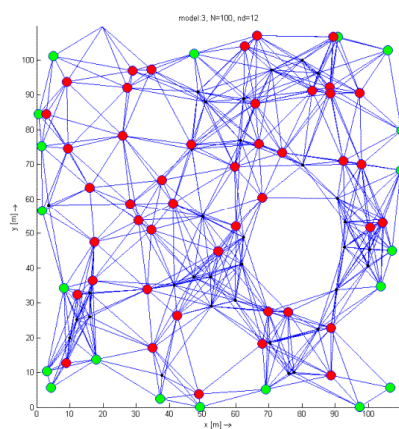


Obr. 17. CGN, BRC, BRB sítě s 50 uzly modely 3. a 9.

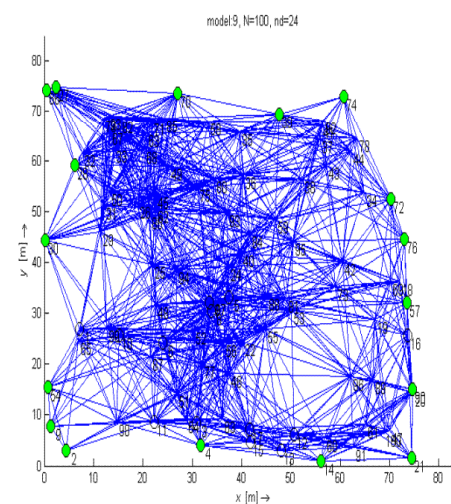
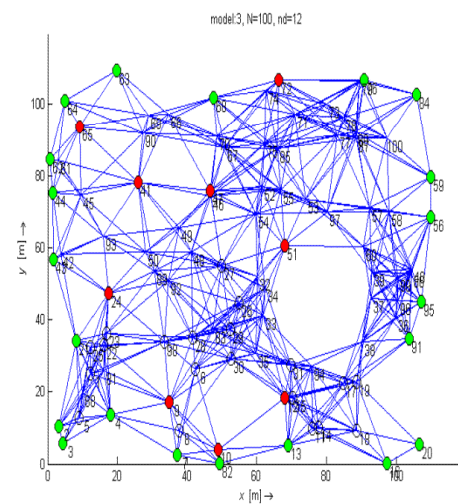
CGN



BRC

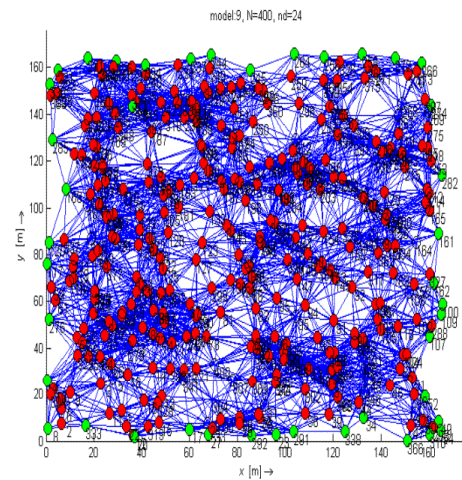
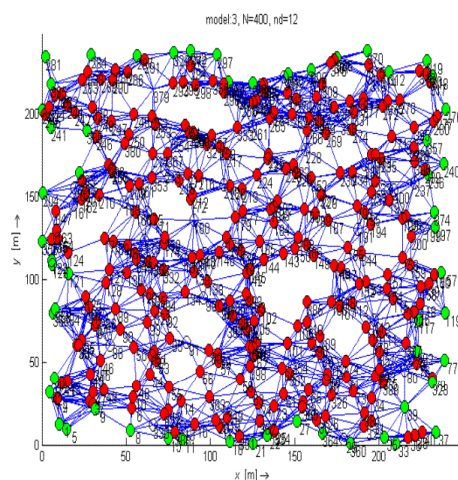


BRB

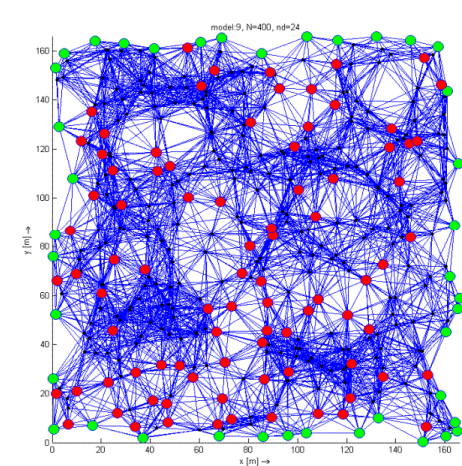
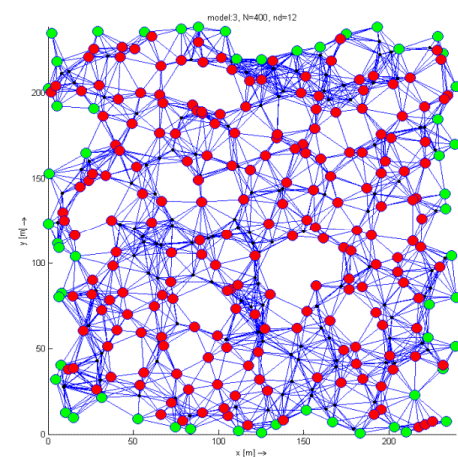


Obr. 18. CGN, BRC, BRB sítě se 100 uzly modely 3. a 9.

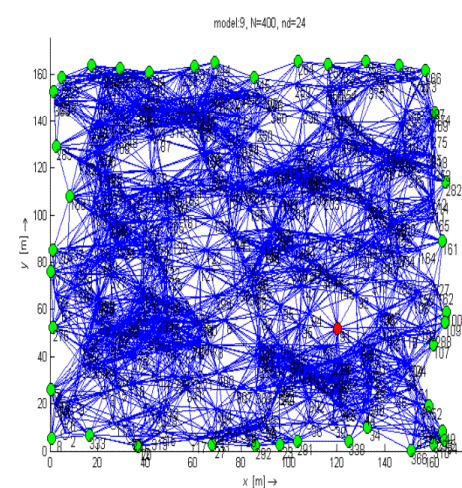
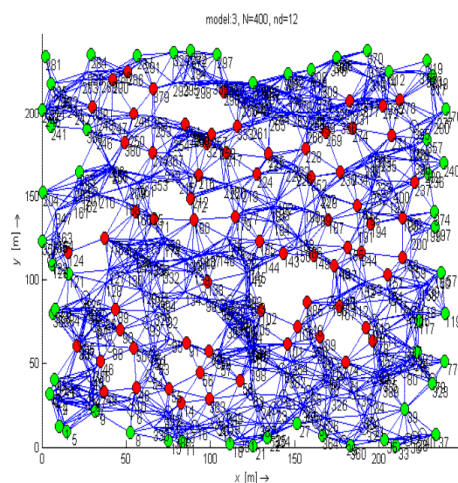
CGN



BRC



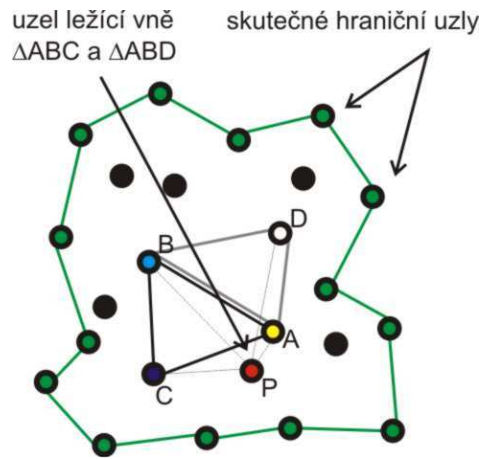
BRB



Obr. 19. CGN, BRC, BRB síť se 400 uzly modely 3. a 9.

U všech grafů na levé straně pro síť s 50 uzly je zřejmé, že u řídkých sítí tzn. model č. 3. je větší množství chybně detekovaných (vyznačeno červeně) hraničních uzlů. Algoritmus CGN určí velké množství uzlů, které detekuje jako hraniční a přitom se nachází uvnitř sítě. Jeden z důvodů velké chybovosti detekce je, že v případě umístění uzlu vně nalezených $\triangle ABC$ a $\triangle ABD$ není ničím zaručeno, že tento uzel a oba trojúhelníky se nachází u okraje sítě. Tato situace je znázorněna na Obr. 20. Proto vylepšením by bylo buďto vhodnějším způsobem vybírat body obou trojúhelníků, nebo zapojit další algoritmus, který by eliminoval chybně označené hraniční body. Algoritmem BRC je vyznačen menší počet chybně detekovaných hraničních uzlů a nejlépe je vyhodnocen algoritmus BRB. Při bližším pohledu na levé spodní grafy (tedy modely č.3) Obr. 17. a Obr. 18. jsou chybně detekovány uzly, kolem nichž neexistuje jediný možný trojúhelník. Další nežádoucí efekt přináší vnitřní hranice sítě viz Obr. 2. a), kde okrajové uzly jsou brány také jako hraniční. Nebudou-li se v síti tyto případy vyskytovat, počet chybně detekovaných uzlů bude menší.

Obdobně je to i u sítí se 100 uzly, kde je pro model č. 3 detekováno větší množství chybných uzlů než pro model 9. U sítě model č. 9 algoritmu BRB nejsou vyznačeny žádné chybné uzly. Horších výsledků dosahují největší síť, tedy se 400 uzly. Algoritmus CGN vyznačil největší množství chybných hraničních uzlů. Výrazně lepších výsledků dosáhl algoritmus BRC a nejlepších BRB. Pro model č. 9. jsou procentuální výsledky chybovosti detekce 76 % (CGN); 18,75 % (BRC); 0,25 % (BRB) viz Tab. 9. Pro největší síť se 400 uzly je algoritmus CGN téměř nepoužitelný vinou vysoké chybovosti. Většinou se podařilo detekovat téměř všechny hraniční uzly, proto úspěšnost detekce dosahuje ve většině případů hodnot větších než 90%.



Obr. 20. Umístění bodu P v síti.

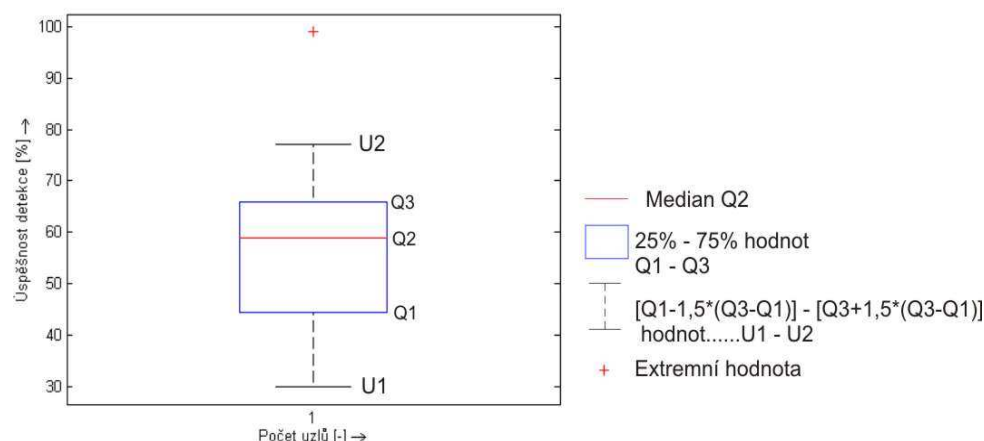
4.3 Úspěšnost a chybovost detekce

Pro vyhodnocení algoritmů CGN, BRC a BRB byla použita úspěšnost a chybovost (redundance) detekce počítány podle vzorců:

$$\text{Úspěšnost detekce} = \frac{\text{Vybr.} \cap \text{Skut.}}{\text{Skut.}} * 100 [\%] \quad (4.1)$$

$$\text{Chybovost detekce} = \frac{\text{Vybr.} - (\text{Vybr.} \cap \text{Skut.})}{\text{numNodes}} * 100 [\%] \quad (4.2)$$

,kde parametr *Vybr.* označuje počet vybraných hraničních uzlů konkrétním algoritmem a parametr *Skut.* vyjadřuje počet skutečných hraničních uzlů dle tabulky (seznam správných hranic). Tedy dalo by se tvrdit, že *Úspěšnost detekce* je podíl počtu skutečných hraničních uzlů vybraných algoritmem k počtu skutečných hraničních uzlů vyjádřen v procentech. *Chybovost detekce* je pak podíl počtu všech chybně vybraných uzlů algoritmem k celkovému počtu uzlů (*numNodes*) vyjádřen v procentech. Tyto dva parametry byly vyneseny do grafů Box plot, jež je součástí Matlabu.



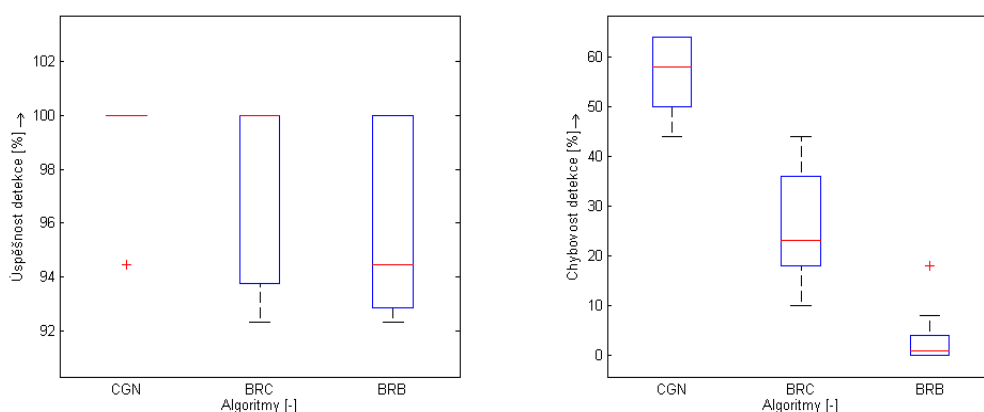
Obr. 21. Příklad Box plotu.

Ilustrační příklad Box plotu je znázorněn na Obr. 21., který je sestaven z hodnot [30, 43, 44, 45, 55, 56, 62, 64, 65, 67, 77, 99]. Většina hodnot je v blízkosti úspěšnosti 60 %, a proto je hodnota mediánu (vyznačena červenou barvou) rovna 59 %. Horní a dolní úroveň (Q3, Q1) modrého obdélníku označuje 25 % až 75 % hodnot. Dolní úsečka pak hodnotu spodní hranice Q1 zmenšenou o jeden a půl násobek šířky Q1 – Q3 a horní hodnotu Q3 zvětšenou o jeden a půl násobek šířky Q1 – Q3, tzn. spodní hranice U1 a horní U2. Křížkem je znázorněn extrém, který se nachází mimo většinu hodnot. Nyní budou porovnány Box ploty úspěšností (na levé straně) a chybovostí (na straně pravé) algoritmů CGN, BRC a BRB tvořeny hodnotami celé simulace. tzn. 10 modelů sítí s 50, 100 a 400 uzly viz Tab. 9.

Č. grafu	Počet uzlů	CGN		BRC		BRB	
		Úspěšnost detekce	Chybovost detekce	Úspěšnost detekce	Chybovost detekce	Úspěšnost detekce	Chybovost detekce
1	50	100	44	100	34	100	18
2	50	94,4444	50	100	40	94,4444	8
3	50	100	58	94,4444	34	94,4444	4
4	50	100	64	100	26	100	2
5	50	100	48	93,75	18	93,75	4
6	50	100	64	100	28	100	0
7	50	100	58	100	18	100	0
8	50	100	58	92,3077	20	92,3077	0
9	50	100	64	92,3077	16	92,3077	0

10	50	100	60	100	10	92,8571	0
11	100	77,1429	56	97,1429	36	94,2857	23
12	100	66,6667	61	100	40	100	14
13	100	95,2381	65	100	43	100	10
14	100	100	65	100	33	100	6
15	100	100	67	100	32	92	4
16	100	100	71	100	26	100	4
17	100	100	69	94,4444	24	88,8889	0
18	100	100	73	100	22	100	0
19	100	100	70	100	19	94,4444	0
20	100	100	73	100	16	94,1176	0
21	400	68	57,25	98	57,75	93	41,75
22	400	88,8889	72,25	97,2222	49,5	95,8333	26,5
23	400	100	73	100	48	96,4912	17,5
24	400	98,3871	73,25	96,7742	42	91,9355	13,5
25	400	91,4894	74,75	93,617	34,5	85,1064	8,5
26	400	93,4783	74	97,8261	34	93,4783	4,5
27	400	97,7778	75	97,7778	26,25	93,3333	2,5
28	400	97,7778	72,5	97,7778	26,25	93,3333	0,75
29	400	97,6744	76	93,0233	18,75	90,6977	0,25
30	400	100	74,5	97,2973	21	94,5946	0,5

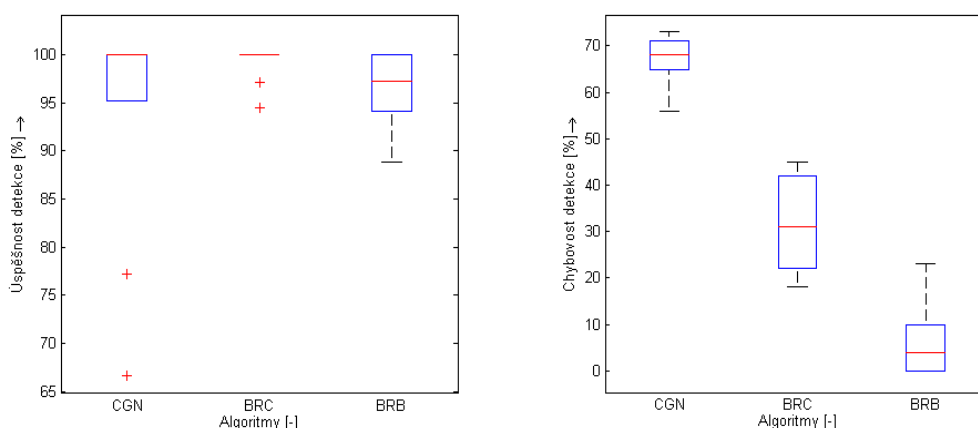
Tab. 9. Tabulka hodnot CGN, BRC, BRB.



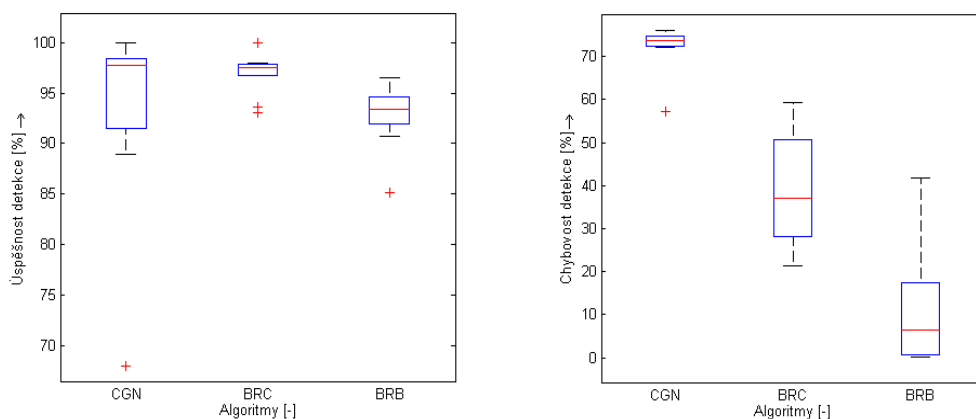
Obr. 22. Úspěšnost a chybovost detekce sítí s 50 uzly.

Hodnota mediánu úspěšnosti detekce znázorněna na Obr. 22., je totožná u algoritmů CGN i BRC. U Box plotu (sítě s 50-ti uzly u CGN) se nezobrazuje obdélník (hranice Q1, Q3)

ani úsečky (U1, U2), protože všechny hodnoty jsou rovny 100 % až na jednu extrémní hodnotu 94,4444 %. Lze tvrdit, že algoritmus BRB dosahuje nízké chybovosti detekce na úkor nižší přesnosti než CGN i BRC. Hodnota mediánu chybovosti detekce algoritmu BRC dosahuje hodnoty kolem 22 % a u algoritmu BRB téměř 0 %. Velkých extrémních hodnot úspěšnosti detekce nabývá CGN u sítí se 100 a 400 uzly, a to pouhých 66,6667 % a 68% viz Obr. 23., Obr. 24. Ze všech tří grafů pro CGN je patrné, že nejmenší hodnota mediánu chybovosti detekce je v síti s 50 uzly, a to až 58 %, což je téměř třikrát větší než hodnota mediánu algoritmu BRC téže sítě. U větších sítí chybovost podstatně roste, což je nežádoucí. Proto lze hodnotit algoritmus CGN jako nejhorší a u velkých sítí nepoužitelný. Důvodem je nevhodný výběr sousedních uzlů a tím i chybné vyhodnocení hraničních uzlů. Pouze u algoritmů BRC i BRB platí, že se zvyšující se hustotou sítě klesá hodnota chybovosti detekce. To je způsobeno tím, že se zvyšující se hustotou sítě roste počet testovaných uzlů a díky tomu lze lépe vybrat vhodné sousední uzly. U algoritmu BRC i BRB platí, že hodnota mediánu chybovosti detekce roste se zvětšujícími se sítěmi (50, 100, 400 uzlů). tzn. pro BRC [23, 31, 38] a pro BRB [1, 5, 7]. Většinou nárůst horní hranice U2 i extrémních hodnot zapříčiňují sítě s řídkým rozmístěním uzlů (tedy modely č. 1. – 3.) z důvodů většího množství vnitřních hranic sítě viz Obr. 2 a). Lze tedy zhodnotit algoritmus BRB jako méně chybový než CGN i BRC, ovšem na úkor nepatrně menších hodnot mediánu úspěšnosti detekce a větší výpočetní náročnosti. Výpočetní náročnosti je podrobněji popsána v kapitole 2.3 Složitost algoritmu.

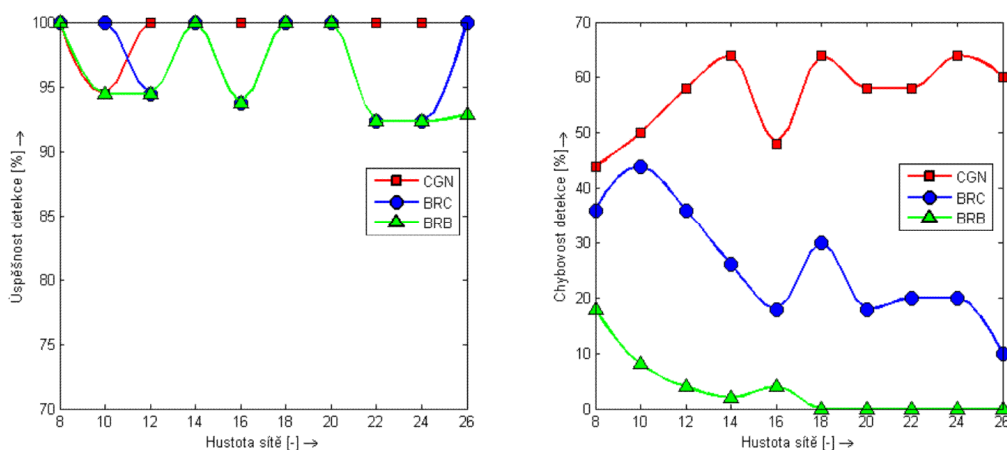


Obr. 23. Úspěšnost a chybovost detekce sítí se 100 uzly.

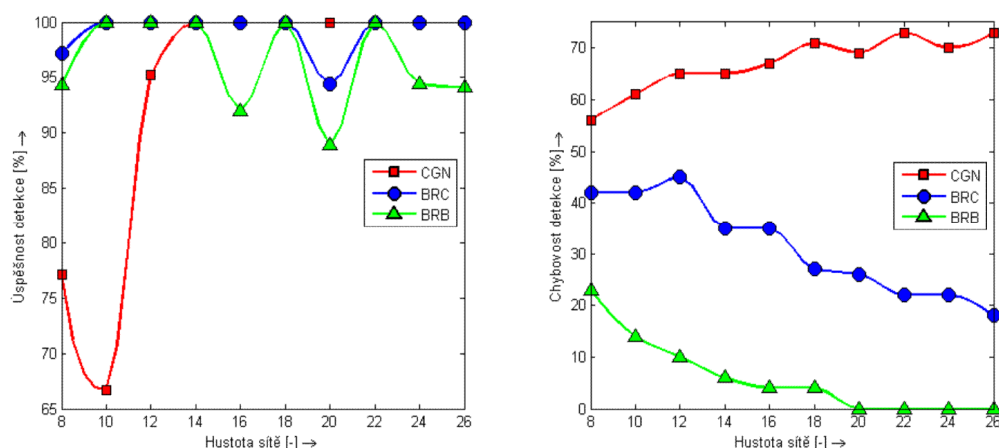


Obr. 24. Úspěšnost a chybovost detekce sítí se 400 uzly.

Na grafech viz Obr. 25., Obr. 26. a Obr. 27. jsou znázorněny závislosti úspěšnosti a chybovosti na hustotě sítě. Z grafu na Obr. 25. je zřejmé, že algoritmus CGN dosáhl nejlepších hodnot úspěšnosti a nezávisí na hustotě sítě. Obráceně pak dosahuje nejvyšších hodnot chybovosti detekce. Algoritmus BRB dosahuje téměř shodných hodnot úspěšnosti jako BRC a výrazně nižší chybovosti detekce. Na Obr. 26. a Obr. 27 pak dosahují hodnoty úspěšností detekce algoritmu BRB horších výsledků než BRC.

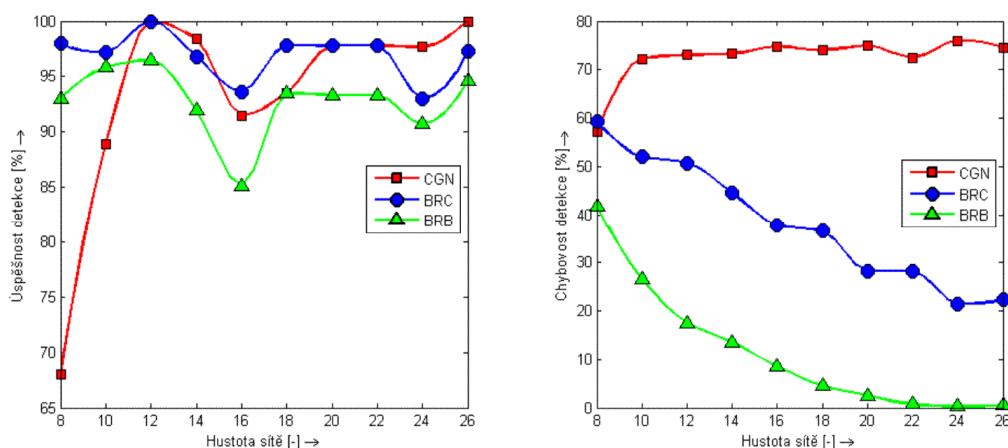


Obr. 25. Úspěšnost a chybovost detekce sítí s 50 uzly v závislosti na hustotě sítě.



Obr. 26. Úspěšnost a chybovost detekce sítí se 100 uzly v závislosti na hustotě sítě.

Pro algoritmy BRC i BRB (Obr. 25., Obr. 26., Obr. 27) platí, že se zvyšující se hustotou sítě roste počet testovaných uzlů a díky lepšímu výběru vhodných sousedních uzlů klesá hodnota chybovosti. Větších výkyvů úspěšnost dosahuje algoritmu CGN u hustoty sítě 8 a 10 viz Obr. 26., Obr. 27.



Obr. 27. Úspěšnost a chybovost detekce sítí se 400 uzly v závislosti na hustotě sítě.

4.4 Zhodnocení algoritmů CGN, BRC a BRB

Všechny tyto algoritmy dosahují vysokých hodnot úspěšností, proto podstatnějším faktem pro zhodnocení bude chybné vyznačení okrajových uzlů.

Jako první bude zhodnocen algoritmus CGN podrobně rozepsán v kapitole

3.1 Algoritmus CGN (Choose Good Neighbors). Základní myšlenka algoritmu vyhledat vhodné sousední body danému uzlu P, a pak určit, zda je uvnitř nebo vně trojúhelníků je inspirativní. Problém ovšem vznikl v reálných náhodných sítích, kde je detekováno velké množství chybných hraničních bodů. Toto velké množství chybných bodů je způsobeno jak výběrem ne zcela ideálních sousedů, tak některými případy popsány v kapitole 3.1.5 Chyby algoritmu CGN. Pro připomenutí, algoritmus CGN si neumí poradit se situacemi na Obr. 6. a), Obr. 7. b), c) a Obr. 8. a), b). Prakticky tím, že není definováno chování pro situaci na Obr. 7. a), nejsou tyto uzly vyhodnoceny jako okrajové, a tedy správně. Obr. 6. b) by byl vyhodnocen správně jako vnitřní.

Extrémní situace lépe řeší algoritmus BRC (viz kapitola 3.2 Algoritmus BRC (Boundary Recognition using Cset)). Dalo by se tvrdit, že BRC je částečně inspirován algoritmem CGN, tedy vhodnějším způsobem vybírá sousední body. Jako poslední bod C jsou vybírány a testovány všechny možné uzly. Tímto jsou eliminovány některé extrémní situace vyznačené na Obr. 7. b), c), ovšem neřeší Obr. 6. a), Obr. 7. a), Obr. 8. a), b). Situace na Obr. 6. b) je vyhodnocena obdobně jako u algoritmu CGN, tedy uzel P je uvnitř $\triangle ABC$. Za zvážení stojí kolik takových případů (Obr. 6. a), b)) může v náhodných sítích nastat, a zda je další řešení velkým přínosem.

Poslední algoritmus BRB (3.3 Algoritmus BRB (Boundary Recognition using Bset)) byl inspirován CGN, BRC a jejich nedostatky. Snaží se najít všechny možné body B i C, a tím dosahuje velmi dobrých výsledků chybovosti detekce viz Obr. 25., Obr. 26., Obr. 27. Skutečností je, že si algoritmus umí poradit se všemi kritickými případy na Obr. 6. b), Obr. 7. b), c) a největším přínosem je řešení situace na Obr. 8. b). Poslední případ se vyskytuje celkem často, a proto jeho vyřešením výrazně klesl počet chybně detekovaných hraničních uzlů, tedy i chybovost detekce. Naopak ale neřeší extrémní situace znázorněny na Obr. 6. a), Obr. 7. a), Obr. 8. a). Za předpokladu, že by se v náhodných sítích příliš nebo vůbec nevyskytovaly případy na Obr. 6. a), lze tuto chybu algoritmu přehlížet. Podstatnější extrémní situace znázorněny na Obr. 7. a) jsou vyhodnoceny jako vnější z důvodu nenalezení všech bodů trojúhelníku. Tyto případy se v testovaných sítích vyskytují několikrát, a proto je nelze přehlížet. Problematické je jak tyto situace odlišit od situací, kdy jsou body opravdu krajní (viz Obr. 7. b), c)). Pouze podmínka je-li v dosahu bodu P alespoň určitý počet sousedních bodů, pak se jedná s velkou pravděpodobností o vnitřní bod, není pro oba případy dostačující. Další dosti častá situace (viz Obr. 8. a)) nastane výběrem nevhodného prvního bodu A. Vyřešit tento problém by bylo možno výběrem více uzlů A a k nim pak zbývající B a C. Výpočetní náročnost

by tím podstatně vzrostla a záleželo by na zaměření a požadavku na čas výpočtu. Zde je rozpor, zda je podstatnější rychlý, i když ne zcela přesný výsledek, nebo přesný výsledek s minimální chybovostí detekce na úkor delšímu času výpočtu. Rozhodnutí záleží již na uživateli a dané aplikaci.

4.4.1 Složitost algoritmu BRB

Nový algoritmus BRB vybírá každému uzlu P jeden uzel A a všechny možné kombinace uzlu B a C (podrobněji viz kapitola 3.3 Algoritmus BRB (Boundary Recognition using Bset)). Tyto kombinace se zapisují do tabulky. Počet řádků tabulky všech možných kombinací trojúhelníků ABC tedy podstatně naroste. V praxi se počet řádků pohybuje pro síť s 50 uzly při dosahu 25 metrů od 1 134 (model č. 1) do 22 230 (model č. 10) a pro síť se 400 uzly od 6 089 (model č. 1) do 126 482 (model č. 10). Na základě počtu řádků lze odvodit složitost větší než kvadratická a menší než kubická tzn. zápis je $O(n^2) < T(n) < O(n^3)$. Pokud není specifikováno jinak, je vhodnější určit asymptotickou složitost (tedy horní hranici složitosti), a pak se téměř vždy jedná o nejhorší možnou složitost. Proto horní hranici složitosti může být $T(n) = O(n^3)$.

Pokus č. 1. Dosah každého uzlu je natolik velký, aby byl schopen komunikovat se všemi uzly v síti s 50 uzly. Počet řádků tabulky všech možných kombinací ΔABC pak naroste do 112 800. Horní hranice složitosti vyhovuje $T(n) = O(n^3)$, jelikož počet prvků vstupních dat je 50 a složitost $T(50) = O(50^3) = 125\,000$. Je zřejmé, že je v praxi toto rozmístění velice nepravděpodobné, avšak je vhodné jej zmínit.

Protože složitost algoritmu BRB je menší než exponenciální (2^n), lze jej považovat za efektivní navzdory větší složitosti než CGN či BRC.

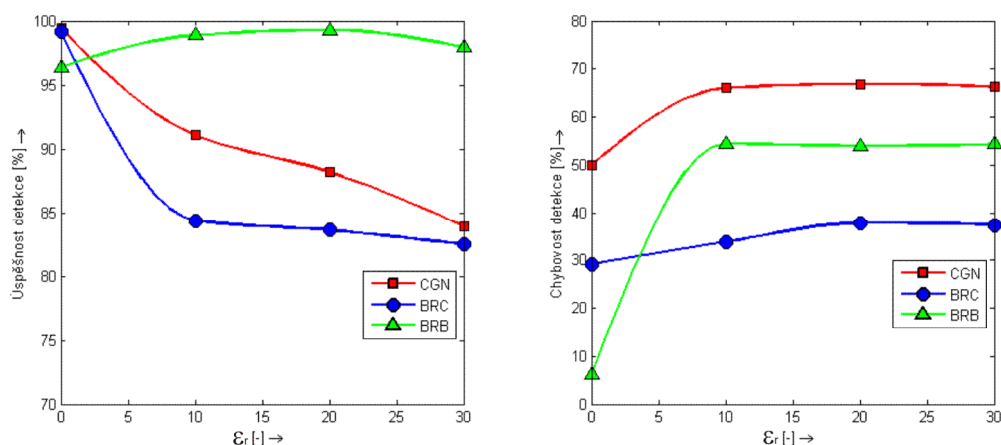
5. Chyby odhadu vzdálenosti

Předchozí sítě byly simulovány v ideálních prostředích při nulové chybě určení vzdálenosti ($\varepsilon_r = 0 \%$). Vzdálenost vyvozená ze síly signálu mezi dvěma uzly byla rovná skutečné vzdálenosti. V reálném prostředí se vyskytují různé fyzikální jevy a ty zapříčiňují rušení bezdrátového signálu. V závislosti na teplotě a objektech v bezdrátovém poli roste rušení signálů i nepřesnost určené vzdáleností. Rušení také způsobuje přítomnost jiného bezdrátového systému (např. WiFi), vysílaného na stejném rádiovém kanálu. Pokud jsou dva uzly od sebe vzdáleny 10 metrů pak při chybě určení vzdálenosti $\varepsilon_r = 10 \%$ jsou vyhodnoceny ze síly signálu ve vzdálenosti 9 nebo 11 metrů. [4]

Parametry simulace	Hodnoty
Databáze vstupních uzlů	networksDB.mat (10 modelů)
Počet uzlů	$numNodes = 100$
Rádiový dosah uzlů	$R = 25 \text{ m}$
Stupeň uzlů	$nd = 8, 10, 12, 14, 16, 18, 20, 22, 24, 26$
Chyba určení vzdálenosti	$\varepsilon_r = 0, 10, 20, 30 \%$ (ideální i reálné prostředí)

Tab. 10. Parametry simulace algoritmů CGN, BRC a BRB, reálné prostředí.

Všechny modely sítí se 100 uzly budou simulovány při chybě určení vzdálenosti $\varepsilon_r = 0, 10, 20$ a 30% viz parametry simulace Tab. 10. Na Obr. 28. jsou znázorněny úspěšnosti a chybovosti detekce v závislosti na chybě odhadu vzdálenosti pro algoritmy CGN, BRC a BRB. Nejlepších hodnot úspěšnosti detekce dosáhl algoritmus BRB a lze tvrdit, že úspěšnost detekce téměř není závislá na zvyšující se chybě odhadu vzdálenosti. Křivky algoritmů CGN a BRC pozvolně klesají pod hodnotu menší než 90% . Algoritmus BRC dosahuje nejhorších výsledků úspěšnosti detekce a zároveň nejlepších výsledků chybovosti detekce. Křivka chybovosti detekce nejprve nepatrně roste do hodnoty $\varepsilon_r = 20 \%$ a pak má konstantní charakter. Horších výsledků chybovosti detekce než algoritmus BRC dosahuje BRB, kde v reálném prostředí ($\varepsilon_r = 10, 20, 30 \%$) nabývá konstantních hodnot kolem 53% . Hodnoty chybovosti detekce BRB jsou výrazně větší v reálném prostředí, než v ideálním prostředí s nulovou chybou odhadu vzdálenosti. V reálném prostředí lze tvrdit, že všechny tři algoritmy dosahují konstantních hodnot chybovosti detekce ($CGN \rightarrow 65 \%$, $BRB \rightarrow 53 \%$, $BRC \rightarrow 38 \%$).



Obr. 28. Úspěšnost a chybovost detekce v závislosti na chybě odhadu vzdálenosti sítě.

Výběr nejvhodnějšího algoritmu v reálném prostředí s rušením bezdrátového signálů závisí na požadavku na velké úspěšnosti detekce hraničních uzlů nebo malé hodnoty chybovosti detekce. Pokud je prioritou úspěšnost detekce při větších chybách odhadu vzdálenosti byl by zvolen algoritmus BRB, ale při požadavku na nízkou chybovost detekce by byl zvolen BRC.

ZÁVĚR

Cílem diplomové práce byl výzkum algoritmů pro detekci hraničních senzorů v bezdrátových sensorových sítích. Byly prostudovány a popsány známé algoritmy CGN, BRC a OBR. Výstupem práce je algoritmus s minimální hodnotou redundance a vysokou úspěšností detekce nazván BRB (Boundary Recognition using Bset). Jelikož všechny tři testované algoritmy (CGN, BRC, BRB) dosahují vysoký počet úspěšně detekovaných hraničních uzlů, bylo podstatnějším hlediskem přesnost detekce tedy minimální hodnota redundance. Samotnému popisu simulačního programu, grafům rozložení úspěšně a chybně detekovaných hraničních uzlů v sítích a porovnání výsledků je věnována čtvrtá kapitola “4. Analýza algoritmů pro detekci hraničních uzlů“. Algoritmus BRB byl porovnán s algoritmy CGN a BRC a jako srovnávací parametry byly použity úspěšnost a chybovost detekce viz např. Obr. 23. Dále byly porovnány jednotlivé grafy závislosti úspěšnosti a chybovosti detekce na hustotě sítě viz Obr. 26. Příklady rozložení uzlů v sítích s úspěšně i chybně vyznačenými hraničními uzly jsou např. na Obr. 18. a zbylé modely sítí jsou přiloženy na DVD. Databáze použitá pro simulaci obsahuje 10 různých modelů s různou hustotou uzlů pro sítě s 50, 100 a 400 uzly. Rádiový dosah je volen 25 metrů pro všechny modely a s rostoucím číslem modelu dochází k zhušťování sítě. Tímto postupem bylo dosaženo 10-ti modelů s různou hustotou sítě. Výsledky simulací tzn. úspěšnost a chybovost detekce jednotlivých algoritmů jsou zapsány v Tab. 9.

V ideálním prostředí dosahuje algoritmus BRB v porovnání s CGN a BRC nejmenších hodnot chybovosti detekce na úkor nepatrně menších hodnot úspěšnosti detekce. Hodnoty mediánu chybovosti detekce algoritmu BRB je většinou přibližně 6x menší než u BRC. Obráceně jsou pak hodnoty mediánu úspěšností detekce algoritmu BRB o 3 – 6 % menší než CGN a BRC. Chybovost detekce algoritmu BRB pozvolna klesá s rostoucí hustotou sítě k nulovým hodnotám. To vypovídá o lepší schopnosti detekovat hraniční uzly v hustších sítích. V reálném prostředí dosahuje algoritmus BRB v porovnání s CGN a BRC nejvyšších hodnot úspěšnosti detekce v závislosti na chybě odhadu vzdálenosti sítě. Nejlepších hodnot chybovosti detekce v závislosti na chybě odhadu vzdálenosti sítě dosahuje algoritmus BRC a o něco horších hodnot BRB. Obecně lze tvrdit, že algoritmus BRB dosáhl v porovnání s CGN i BRC nejlepších výsledků kvůli nejmenší redundanci a nepatrně horších hodnot úspěšnosti detekce. Negativem BRB je větší výpočetní náročnost a horší hodnoty chybovosti detekce v závislosti

na chybě odhadu vzdálenosti sítě. Nejhorších výsledků dosáhl algoritmus CGN kvůli velkému počtu chybně detekovaných hraničních uzlů.

SEZNAM LITERATURY

- [1] Swami A., Zhao Q., Hong Y., Tong L.: “Wireless Sensor Networks, Signal Processing and Communications Perspectives.” , ISBN: 978-0-0470-03557-3, 2007.
- [2] Deogun J.S. a kol.: “An Algorithm for Boundary Discovery in Wireless Sensor Networks.” University of Nebraska-Lincoln, USA 2005.
- [3] Saukh O., Sauter R. a kol.: “On Boundary Recognition without Location Information in Wireless Sensor Networks. ” University Bonn and Fraunhofer IAIS, University Stuttgart, Germany.
- [4] Šimek M.: “Reference Nodes selection for anchor-free localization in wireless sensor network.” Vysoké učení technické v Brně, ČR 2010.
- [5] Mao G., Fidan B., a Anderson B. D. O., “Wireless Sensor Network Localization Techniques”, Comput. Netw., vol. 51, no. 10, pp. 2529–2553, 2007.
- [6] Priyantha N. B., Balakrishnan H., Demaine E. D., a Teller S. J., “Anchor-Free Distributed Localization in Sensor Networks”, v SenSys, pp. 340-341, 2003.
- [7] Hoffmann-Wellenhof B., Lichtenegger H. a Collins J., “Global Positioning System: Theory and Practice: Fourth Edition.” Springer-Verlag, 1997.
- [8] Koubková A., Pavelka J., “Úvod do teoretické informatiky”, vyd. 3, Praha: Matfyzpress, ISBN 80-86732-03-7.
- [9] Burget R., “Hodnocení algoritmů”, Přednáška MTIN, Vysoké učení technické v Brně, ČR 2010.

[10] Burget R., “Abstraktní datové typy”, Přednáška MTIN, Vysoké učení technické v Brně, ČR 2010.

[11] Burget R., “Vyčíslitelnost a složitost P, NP a NPC”, Přednáška MTIN, Vysoké učení technické v Brně, ČR 2010.

[12] Burget R., “Řadící algoritmy”, Přednáška MTIN, Vysoké učení technické v Brně, ČR 2010.

[13] Römer K., Friedermann Mattern, “The Design Space of Wireless Sensor Networks”, Institute for Pervasive COmputing ETH Zurich, 2004.

SEZNAM ZKRATEK, VELIČIN A SYMBOLŮ

2D	Dvoj dimenzionální
a [-]	Konstanta pro lineární časovou složitost
ADT	Abstraktní datový typ
AFL	Anchor-Free Localization
b [-]	Konstanta pro lineární časovou složitost
BRB	Boundary Recognition using Bset
BRC	Boundary Recognition using Cset
c [-]	Konstanta pro exponenciální třídu složitosti
C_0 [-]	Cyklus nebo taky chordless cyklus
CGN	Choose Good Neighbors
d [m]	Minimální vzdálenost uzlů
DFT	Discrete Fourier transform
d_{PA} [m]	Distance P, A
DVD	Digital Versatile Disc
E [-]	Množina neorientovaných hran
ε_r [%]	Chyba určení vzdálenosti
F	Oblast
FFT	Fast Fourier transform
G	Graf
GPS	Global Position Systém
h [-]	Počet přeskoků
Chybovost detekce [%]	
i	Index uzlu
ID	Identification Number
IP	Interior Point
j	Index uzlu
min_C [-]	Minimální konečná oblast
n [-]	Velikost vstupních dat
$N(P)$	Seznam sousedních uzlů k uzlu P
nd [-]	Stupeň uzlů
NP	Nondeterministic polynomial time
$numNodes$ [-]	Počet uzlů (50, 100, 400)
$O(1)$	Notace Omikron

OBR	On boundary recognition without location informatik
P	Počáteční uzel
R [m]	Rádiový dosah (25 m)
S	Soubor uzlů A, B, C nebo u OBR význam jádro S.
$S_{A,B,C}$ [m ²]	Obsah plochy A, B, C
$Skut.$ [-]	Počet skutečných hraničních uzlů
$T(n)$	Časová složitost s n vstupními daty
Úspěšnost detekce [%]	
V [-]	Soubor uzlů
$Vybr.$ [-]	Počet vybraných hraničních uzlů algoritmem
WiFi	Wireless LAN
WSN	Wireless Sensor Network

PŘÍLOHA

Příloha 1.: Obsah DVD

- Programy pro detekci hraničních uzlů algoritmů CGN, BRC a BRB.
- Databáze uzlů a seznam hraničních uzlů.
- Tabulky, grafy a obrázky rozmístění uzlů v sítích.
- Elektronická verze tohoto dokumentu.